

KNOWLEDGE DISTILLATION AS A SOLUTION FOR DOMAIN  
TRANSFER

by

Mithun Paul

---

Copyright © Mithun Paul 2022

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF COMPUTER SCIENCE

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2022

THE UNIVERSITY OF ARIZONA  
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by: Mithun Paul, titled: *Knowledge Distillation as a Solution for Domain Transfer*.

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

*Mihai Surdeanu*  
Date: Apr 15, 2022

*Mihai Surdeanu, Committee Chair*

*J. Barnard*  
Date: Apr 14, 2022

*Jacobus Barnard, Committee Member*

*Clayton*  
Date: Apr 14, 2022

*Clayton Morrison, Committee Member*

*Sue Brown*  
Date: Apr 14, 2022

*Sue Brown, Committee Member*

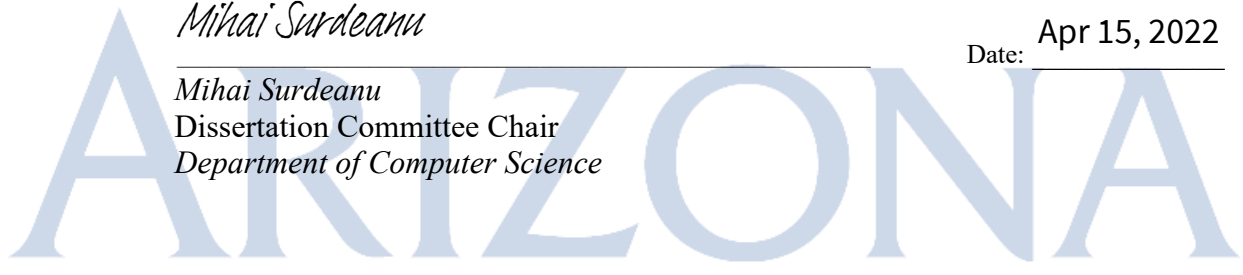
Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.



*Mihai Surdeanu*  
Date: Apr 15, 2022

*Mihai Surdeanu*  
Dissertation Committee Chair  
Department of Computer Science



## ACKNOWLEDGEMENTS

First, I would like to thank a few amazing geniuses without whose spark of ideas this work wouldn't have existed. First, Mihai for his constant critiques. I also want to thank my Committee members, Dr. Kobus Barnard, Dr. Clayton Morrison and Dr. Susan Brown. Next I want to thank my ex-wife Deepa for putting up with everything. Thank You. You were the best thing that ever happened to me. I also want to thank my colleague, collaborator and close friend, Sandeep Suntwal, whose initial curiosity on semantic patterns helped us discovering the problem on which this dissertation is based on. Without him, his constant cheerful demeanor and support, this dissertation wouldn't have taken off, or sustained this far. Next is Becky Sharp; de-lexicalization as a solution for this problem was a flash of genius from Becky. She once mentioned 'neutering' as a possible solution 'en-passe' in a 'random' brain storming meeting - and still refuses to take credit for it. Steve Bethard has been a constant sounding board for my ideas and grievances. And I am yet to find a better teacher than him in 32 years of my academic life. Also, I want to thank Ajay Nagesh, who using his vast repository of knowledge accrued from his voracious appetite for reading of research papers, introduced me to the teacher student architecture work. And then Keith and Marco, for being the scapegoats of my stupid programming issues - and holding my hand always patiently through them. Others I must mention earnestly for their support are everyone in my lab, including Pooja, George, Masha, Andrew, Robert, Hoang, Shariar, Enfa, Divija et al. Next my close friends Nithin, Harshad, Faith, Aurora and our lovely dog, Callista, whose constant presence with licks and drools even makes her a co-author of this dissertation. This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the Habitus program, and by the Bill and Melinda Gates Foundation HBGDki Initiative across years. If I have missed mentioning someone it is purely due to my meager memory and not intentional at all.

## DEDICATION

To Deepa. The light that was. And then wasn't.  
Sometimes you have to sacrifice your queen to get to the end game.

## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	8
LIST OF TABLES . . . . .	10
ABSTRACT . . . . .	15
CHAPTER 1 INTRODUCTION . . . . .	17
1.1 Need for domain transferable learning . . . . .	18
1.2 Neural Network Overfitting On Patterns . . . . .	19
1.3 Contributions . . . . .	20
1.3.1 Delexicalizing fact verification datasets to improve domain transfer . . . . .	20
1.3.2 Creating a neural network based method to control over delex- icalization . . . . .	21
1.4 Creating an architecture to automatically arrive at the right amount of delexicalization needed . . . . .	22
1.5 Overview . . . . .	23
CHAPTER 2 RELATED WORK . . . . .	24
2.1 Natural Language Inference . . . . .	24
2.1.1 Datasets . . . . .	25
2.2 Recognizing Textual Entailment . . . . .	26
2.2.1 Datasets . . . . .	26
2.2.2 Methods for Recognizing Textual Entailment . . . . .	28
2.3 Fact Verification . . . . .	34
2.3.1 Datasets . . . . .	34
2.3.2 Methods . . . . .	41
2.4 Domain Adaptation . . . . .	43
2.4.1 Methods that modify models . . . . .	46
2.4.2 Data modification based methods . . . . .	51
2.4.3 Methods that use both data centric and model centric ap- proaches . . . . .	54

TABLE OF CONTENTS – *Continued*

CHAPTER 3	Towards the Necessity for Debiasing Natural Language Inference	
	Datasets . . . . .	56
3.1	Masking Techniques . . . . .	58
	3.1.1 Overlap-Aware Named Entity Recognition . . . . .	58
	3.1.2 OA-NER + Super Sense (SS) Tags . . . . .	59
3.2	Datasets and Methods . . . . .	60
3.3	Results and Discussion . . . . .	61
3.4	Qualitative Analysis . . . . .	63
	3.4.1 Positive Changes . . . . .	64
	3.4.2 Negative Changes . . . . .	65
3.5	Conclusion . . . . .	66
3.6	Language Resource References . . . . .	67
CHAPTER 4	Using Data Distillation To Achieve Domain Transfer . . . . .	68
4.1	Need for domain transfer in fact verification . . . . .	68
4.2	Overview . . . . .	69
4.3	Setup . . . . .	70
4.4	Datasets and models . . . . .	73
	4.4.1 Datasets . . . . .	74
	4.4.2 Models . . . . .	75
4.5	Attention analysis . . . . .	78
4.6	Delexicalization . . . . .	79
	4.6.1 Named Entity Recognition Tools . . . . .	80
	4.6.2 Masking Techniques . . . . .	82
4.7	Results . . . . .	85
4.8	Discussion . . . . .	86
	4.8.1 Delexicalization learns and preserves . . . . .	86
	4.8.2 Delexicalization learns and improves . . . . .	87
	4.8.3 Delexicalization learns and improves in cross-domain setting . . . . .	88
	4.8.4 Percentages . . . . .	88
	4.8.5 Drop in performance with super sense tags . . . . .	89
4.9	Conclusion . . . . .	90
CHAPTER 5	Data and Model Distillation as a Solution for	
	Domain-transferable Fact Verification . . . . .	94
5.1	Overview . . . . .	95
5.2	Related Work . . . . .	95
5.3	Data distillation . . . . .	98
	5.3.1 FIGER . . . . .	99
	5.3.2 De-lexicalization with FIGER . . . . .	100

TABLE OF CONTENTS – *Continued*

5.4	Model distillation . . . . .	101
5.5	Models . . . . .	103
5.6	Experiments . . . . .	104
5.7	Results . . . . .	105
5.8	Error Analysis . . . . .	106
5.9	Conclusion . . . . .	107
CHAPTER 6 Group Learning: A Collective Knowledge Distillation Solution for Domain Transfer in Fact Verification . . . . . 108		
6.1	Data Distillation . . . . .	109
6.2	Model Distillation . . . . .	110
6.3	Models and Data . . . . .	111
	6.3.1 Models . . . . .	111
	6.3.2 Data . . . . .	112
6.4	Experiments . . . . .	112
6.5	Results . . . . .	112
6.6	Discussion . . . . .	113
	6.6.1 Which de-lexicalization technique is the best. . . . .	113
	6.6.2 Higher attention to tokens across claim-evidence boundaries. . . . .	113
6.7	Conclusion . . . . .	117
APPENDIX . . . . .		125
REFERENCES . . . . .		128

## LIST OF FIGURES

4.1	Experiment setup for testing the hypothesis that state of the art models drop in performance when tested on a dataset it was not trained on. In our experiments we first train on the training partition of one dataset (e.g.,FEVER), and use the same trained model to predict and evaluate on the test partition of the same dataset (e.g.,FEVER). The accuracy achieved here is considered as in-domain performance. We also use the same model to test on the test partition of the other dataset (e.g.,FNC). The accuracy achieved here is considered as the cross-domain performance of the corresponding model. Then all these experiments are run in the other direction i.e with the datasets swapped from first experiment. . . . .	73
4.2	Pictorial overview of the Decomposable Attention approach, showing the Attend (left), Compare (center) and Aggregate (right) steps. Specifically, F denotes the a function in the attend step using which the unnormalized attention weights are obtained. . . . .	76
4.3	Distribution of POS tags that were assigned the highest attention weights by DA for incorrectly classified cross-domain examples. . . . .	78
4.4	Examples of super sense categories for nouns and verbs . . . . .	82
5.1	The mean teacher architecture. The figure depicts a training batch with a single labeled example. Both the student and the teacher model evaluate the input applying noise within their computation. The softmax output of the student model is compared with the one-hot label using classification cost and with the teacher output using consistency cost. After the weights of the student model have been updated with gradient descent, the teacher model weights are updated as an exponential moving average of the student weights. Both model outputs can be used for prediction, but at the end of the training the teacher prediction is more likely to be correct. A training step with an unlabeled example would be similar, except no classification cost would be applied. . . . .	97
5.2	112 tags used in FIGER. The bold-faced tag is a rough summary of each box. The box at the bottom right corner contains mixed tags that are hard to be categorized. . . . .	98
5.3	The teacher-student architecture for model distillation. . . . .	101

LIST OF FIGURES – *Continued*

6.1	The multiple-student architecture for knowledge distillation. . . . .	110
6.2	Weights assigned by the attention heads in the first layer of BERT for a given claim-evidence pair when predicting using a student model trained on plain-text data. It can be seen that despite multiple tokens available from the given claim-evidence pair, most attention weight is assigned to the BERT specific tokens like [CLS] and [SEP]. . . . .	114
6.3	Weights assigned by the attention heads in the last layer of BERT for a given claim-evidence pair when predicting using a student model trained on plain-text data. It can be seen that despite multiple tokens available from the given claim-evidence pair, most attention weight is assigned to the BERT specific tokens like [CLS] and [SEP]. . . . .	115
6.4	The percentage of all attention weights for all the claim-evidence pairs in the fact verification dataset using a student model that was trained using plain-text data. For all the claims in the FEVER dataset, 62.1% of all attention weights came from tokens in the same claim text itself, while only 37.9% came from the corresponding evidence text. . . . .	116
6.5	The percentage of all attention weights for all the claim-evidence pairs in the fact verification dataset using a student model that was trained using de-lexicalized data. For all the claims in the FEVER dataset, only 37.5% of all attention weights come from tokens in the same claim text itself, while 62.5% come from the corresponding evidence text. . . . .	117

## LIST OF TABLES

3.1	Example illustrating our masking techniques, compared to the original fully lexicalized data. . . . .	58
3.2	Performance of various high performing NN methods over lexicalized and de-lexicalized versions of the same dataset. ‘Matched’ is the in-domain partition of the MNLI validation dataset, and ‘mis-matched’ is the out-of-domain partition. The performance of both the methods remain close to each other in de-lexicalized and lexicalized versions of the same dataset, which validates that our de-lexicalization techniques preserve the original information of the text. . . . .	59
3.3	Performance accuracies of the Decomposable Attention against various masking techniques when tested out-of-domain. The accuracies reported are the mean across 3 random seed experiments using the Decomposable Attention model, with their corresponding standard deviations mentioned in brackets. The “Train Domain” row indicates the training datasets, while the “Eval Domain” indicates the domain of the corresponding evaluation partitions. For example, one experiment trained the DA method on FEVER and evaluated the resulting model on the testing partition of FNC (column 3). . . . .	60
3.4	Performance accuracies of the Decomposable Attention against various masking techniques when tested in-domain for FNC and FEVER datasets. The “Lexicalized” row shows the accuracies when DA was trained using the corresponding lexicalized data. This demonstrates that while de-lexicalization with OA-NER maintains the performance, the addition of Super Sense tags reduces the accuracy, emphasizing the fact that the amount of granularity to use is still an open problem. The accuracies reported are the mean across 3 random seed experiments using the Decomposable Attention model, with their corresponding standard deviations mentioned in brackets. . . . .	61
3.5	Data points in which the model trained on lexicalized data made an incorrect prediction while the model trained on the data de-lexicalized with OA-NER made the right prediction. . . . .	63
3.6	The percentage of labels that were mis-classified by the model that was trained on de-lexicalized data. . . . .	63
3.7	Confusion matrix that highlights the distribution of the incorrectly predicted labels. Rows indicate the incorrectly predicted labels and columns indicate the corresponding original correct labels. . . . .	64

LIST OF TABLES – *Continued*

3.8	An example of a data point whose original label was <i>discuss</i> and the model trained on overlap-aware de-lexicalized data predicted as <i>agree</i> .	65
3.9	Relative percentages of some OA-NER tags with respect to their labels along with the original label distribution in the training data. .....	65
4.1	Performance of a state of the art model, BERT, that was trained in one domain, drops considerably when tested on a related domain. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets. ....	69
4.2	Examples of claim and evidence sentences from FEVER dataset (Thorne et al., 2018a) .....	71
4.3	Examples of claim and evidence sentences from FNC dataset (Pomerleau and Rao, 2017). Refer appendix for full text of evidence sentences. ....	72
4.4	Label distribution for the FEVER and FNC datasets. We consider the <i>agree</i> and <i>disagree</i> FNC labels as equivalent to the <i>support</i> and <i>refute</i> labels in FEVER. The FNC <i>not enough info (NEI)</i> label is listed below the more fine-grained <i>discuss</i> and <i>unrelated</i> FEVER labels. .	74
4.5	An example of the named entities as found by the Stanford CORENLP named entity recognizer. ....	80
4.6	An example of identifying and labeling several lexical expressions with supersenses (Schneider and Smith, 2015) in a given sentence. Note that uppercase is used to denote nouns, while lowercase is used to denote verbs. ....	81
4.7	Example illustrating our various masking techniques, compared to the original fully lexicalized data (first row). Note that the masking tags were generated with real-world (imperfect) tools. For example, “Airbus A380” in the claim was correctly classified as <b>miscellaneous</b> by the NER tool, while “A380” in the evidence was not, thus preventing us from taking advantage of the overlap. ....	83
4.8	Various masking techniques and their performance accuracies, both in-domain and out-of-domain. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets. ....	85

LIST OF TABLES – *Continued*

- 4.9 An example of prediction on a claim-evidence pair from the FEVER dataset to show that using delexicalization learns and preserves knowledge. Each of the rows show the predictions made on the same claim-evidence pair by 2 models: one which was trained on lexicalized/plain-text dataset, and the other which was trained on the same dataset which was delexicalized with the OA-NER technique. In both cases, the models predict the labels correctly. In case of the model trained on lexicalized data, as shown in the 6th column, the word ‘Stiller’ is considered as having very high importance. However when using the model trained on delexicalized data, the word ‘Stiller’ doesn’t exist in the list of words with highest importance/attention as given by the model. This is proof that delexicalization learns and preserves knowledge. . . . . 91
- 4.10 An example of a claim-evidence pair from FEVER, with the corresponding prediction from the DA model, with and without delexicalization. In the first row, we show that the lexicalized model predicts the label to be *agree* despite the actual label being *disagree*. However, as seen in the second row, when the model was trained on delexicalized data, it was able to predict the correct label, *disagree*. Further, as shown in the 6th column, the word ‘Stiller’ is not present in the top 3 words with highest attention anymore when the model trained on delexicalized data was used for prediction. This shows that delexicalization learns the underlying semantics and also improves the prediction. . . . . 92
- 4.11 An example of how delexicalization improves prediction in the cross domain setting. Like before, both rows show the predictions made by 2 DA models on an example from the FNC dataset Pomerleau and Rao (2017). However, the difference is that these two models were trained on the FEVER dataset. The first row is the prediction by the model which was trained on the lexicalized version of the FEVER dataset. The second row is the prediction of the model which was trained on the delexicalized version of the FEVER dataset. As seen from the 5th column, the model trained on lexicalized FEVER data predicts the label of the FNC claim-evidence pair incorrectly. However, the model trained on the FEVER data delexicalized using OA-NER technique, predicts the label correctly. Further as seen in the 6th column, the words with the highest attention included ‘Stiller’, while in the second row it doesn’t. . . . . 93

LIST OF TABLES – *Continued*

5.1	An example of identifying and labeling several lexical expressions with FIGER (Ling and Weld, 2012) in a given sentence. . . . .	99
5.2	An example of a claim-evidence pair from the FEVER dataset (Thorne et al., 2018b), de-lexicalized using fine-grained tags from FIGER (Ling and Weld, 2012). . . . .	100
5.3	In-domain and cross-domain accuracies for various methods. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets. “BERT Lex” is the stand alone model trained on the original lexicalized data; “BERT Delex” is the standalone model trained on de-lexicalized data. OA-NER de-lexicalizes the data using the Overlap Aware Named Entity Recognizer; SS uses Super Sense tags — two de-lexicalization techniques mentioned in Sun et al. (2019a). FIGER de-lexicalizes the data using a fine-grained named entity recognizer (Ling and Weld, 2012). ; and “BERT TS” denotes the student in the proposed teacher-student architecture. * indicates that the corresponding result is significantly better than its baseline (“BERT lex” in the same column), under a bootstrap resampling test with 1,000 samples, and $p$ -value $< 0.035$ . . . . .	104
5.4	Top 10 tokens with the highest attention weights by each of the trained models. ‘Lex’ is the stand alone model trained on the original lexicalized data and ‘TS Student’ denotes the student in the proposed teacher-student architecture. . . . .	106
6.1	Examples of a claim-evidence pair of sentences used in fact verification being transformed using various delexicalization techniques. The delexicalization labels range from more abstract to more specific. For example the first row shows the given claim evidence pair in its original plain text format. In the second row the same text is delexicalized by replacing some concepts with their corresponding named entities identified using the Stanford named entity recognizer (Finkel and Manning, 2009). The third row shows the same data delexicalized with an entity recognizer from AllenAI (Ling and Weld, 2012) and a set of abstract types. The fourth row uses the same entity recognizer, but with more specific, i.e., more fine-grained, entity types. For example, we now replace <i>Tolkien</i> with <b>author</b> rather than <b>person</b> . . . . .	119

LIST OF TABLES – *Continued*

6.2	In-domain and cross-domain accuracies for various methods. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets. “Lex” is the stand alone model trained on the original lexicalized data; “GL” denotes the student in the proposed multi-student ‘Group Learning’ architecture. * indicates that the corresponding result is significantly better than its baseline (“lex” in the same column), under a bootstrap resampling test with 1,000 samples, and $p$ -value $< 0.05$ . . . . .	120
-----	---	-----

## ABSTRACT

While neural networks produce state-of-the-art performance in several NLP tasks, they generally depend heavily on lexicalized information, which transfer poorly between domains. In this dissertation we investigate this problem deeply and present various solutions. First we investigate the importance that a model assigns to various aspects of data while learning and making predictions, specifically, in a fact verification task. By inspecting the attention weights assigned by the model, we confirm that most of the weights are assigned to noun phrases. This reaffirms our initial hypothesis that neural network based models depend heavily on lexicalized information for their learning. Next we propose a solution to mitigate this dependence on lexicalized information. This solution, called knowledge distillation, is a combination of two strategies, data distillation and model distillation. The first one, data distillation, is a myriad of data masking techniques which enables delexicalization. We show that by using our solution, not only does the performance of an existing state-of-the-art model remain at par with that of the model trained on a fully lexicalized data, but it also performs better than it when tested out of domain. Further to encourage models to extract transferable facts from a given fact verification dataset, we combine the data distillation techniques with a model distillation technique based on the teacher-student architecture (Tarvainen and Valpola, 2017). Results show that the performance is further improved in an out-of-domain setting. However, that solution does not address the critical issue of how much delexicalization should be applied? For example a little helps reduce over-fitting, but too much discards useful information. To resolve that, we propose Group Learning (GL), a model distillation approach for fact verification. In our method, while multiple student models have access to different delexicalized data views, they are

encouraged to independently learn from each other through pair-wise consistency losses. In several cross-domain experiments between the FEVER and FNC fact verification datasets, we show that our approach learns the best delexicalization strategy for the given training dataset and outperforms state-of-the-art classifiers that rely on the original data. For example, the performance of a state-of-the-art RTE model trained on the Fact Extraction and Verification (Thorne et al., 2018) dataset and evaluated on Fake News Challenge (Pomerleau and Rao, 2017) improved by over 10% in accuracy score compared to the fully lexicalized model.

## CHAPTER 1

### INTRODUCTION

The ever-increasing proliferation of online text means that information is available on essentially any topic. However, it has become increasingly difficult to distinguish between fake and real information. For example, in a recent survey around 67% Americans reported that fake news causes confusion about simple information that they read online and 54% say that fake news affects their confidence in other people they interact with, making fake news a very important societal issue. Clearly, ever-improving tools are needed to handle this massive (and growing) abundance of fake information, and the vast majority of this data is in the form of natural language. While certain fake information texts can be classified using statistics, in order to handle a new fake information text, the system must be able to compare that information with other reliable knowledge sources and *infer* what the desired result is from the given text.

Specifically, in the natural language processing world, this problem is called *fact verification* which is considered a sub task of the natural language inference task. NLI is the task of determining if one piece of text can be plausibly inferred from another. The first piece of text is typically called a claim or hypothesis while the second piece of text is called an evidence or a premise. Thus given a pair of claim and evidence sentences, the task then is to classify them into 3 classes, depending on whether they **agree**, **disagree** or are **neutral** to each other.

Simply put NLI incorporates natural language understanding capabilities in a very simple interface: detecting when the meaning of one piece of text is contained in the meaning of another. This straightforward representation of a difficult problem appeals to a wide range of people, in part because it can be used as a component in a variety of different NLP applications, such as Machine Translation, Semantic

Search, and Information Extraction. It also avoids committing to a certain meaning representation and reasoning paradigm, allowing it to appeal to a broader range of researchers.

However, the key differences between fact verification task and NLI are as follows: In NLI systems the evidence to verify each claim is given, whereas in fact verification systems, the passage is typically retrieved from a large set of documents (e.g., Wikipedia or news articles) in order to form the evidence. Also in NLI both the passages to verify has typically consisted of a single sentence while in fact verification the evidence is usually longer passages. Also the labels of the classification tasks are also different. For example, in the 2018 Fact Extraction and Verification (FEVER) challenge (Thorne et al., 2018a), the NLI module was used determine if a given set of evidence sentences, when compared with the claim provided, can be classified into either of **supports**, **refutes**, or **not enough information**.

In this work we tackle the fake information problem as a fact verification task.

to achieve that it is important that the datasets and models used for this purpose need to be made robust by creating neural network methods which can learn knowledge that is domain transferable. Next we elaborate on why this is important.

## 1.1 Need for domain transferable learning

While there have been several methods created to address and automate online fact verification, humans still seem to have a better ability than machines to distinguish good facts from bad facts. We<sup>1</sup> believe this is because we use background and domain knowledge while these methods only look at facts as independent silos. Inspired from this observation, my research aims on building robust methods for fact verification by aggregating information across multiple fact datasets (e.g., Wikipedia based datasets or news articles based ones). To thus achieve information aggregation for fact verification, it is important that the neural network methods used for training

---

<sup>1</sup>Unless otherwise specified, plural first-person pronouns refer to my collaborators and I, where collaborators come mostly from the computational language understanding (CLU) lab at the University of Arizona.

over these facts be made robust by having them learn knowledge that is transferable from one domain to another.

To ensure robustness in a task, not only should the neural network learning methods used be domain transferable, but the corresponding datasets also must be bias free. Next we discuss why that is important.

## 1.2 Neural Network Overfitting On Patterns

In order to advance any task in natural language processing, quality data sets are quintessential. Some such data sets which have enabled the advancement of NLI (and fact verification) are SNLI (Bowman et al., 2015) MNLI (Williams et al., 2017), FEVER (Thorne et al., 2018a), and FNC (Pomerleau and Rao, 2017). However these datasets are not devoid of biases (subtle statistical patterns in a dataset, which could have been introduced either due to the methodology of data collection or due to an inherent social bias). For example, (Gururangan et al., 2018b) and (Poliak et al., 2018) show that biases were introduced into the MNLI dataset by certain language creation choices made by the crowd workers. Similarly, (Schuster et al., 2019) show that in the 2018 Fact Extraction and Verification (FEVER) challenge (Thorne et al., 2018a), the `refutes` label highly correlates with the presence of negation phrases.

These biases can be readily exploited by neural networks (NNs), and thus have influence on performance. As an example, (Gururangan et al., 2018b) demonstrate that many state of the art methods in NLI could still achieve reasonable accuracies when trained with the hypothesis alone. Similarly, (Suntwal et al., 2019b) show that some NN methods in NLI with very high performance accuracies are heavily dependent on lexical information. Further (Yadav et al., 2019) show that this issue is relevant not just in NLI but in other NLP applications such as question answering (QA) also.

This tendency of NNs to inadvertantly exploit such dataset artifacts is likely worsened by the fact that currently the success of NLP approaches is almost exclusively measured by empirical performance on benchmark datasets. While this emphasis on performance has facilitated the development of practical solutions, they

may lack guidance as they are often not motivated by more general linguistic principles or human intuition. This makes it difficult to accurately judge the degree to which these methods actually extract reasonable representations, correlate with human intuition or understand the underlying semantics (Dagan et al., 2013). Thus in order to build a robust system for fact verification, the first step is to find quality datasets and ensure that they are devoid of biases.

### 1.3 Contributions

With this work we tackle the challenging task of fact verification, seeking methods which balance the demands of robustness, accuracy and transferability. We address these challenges with three approaches. In particular, the specific contributions of this work are:

#### 1.3.1 Delexicalizing fact verification datasets to improve domain transfer

First we inspect and confirm that neural networks depend heavily on certain statistical nuances and lexical artifacts found in datasets. We do that by inspecting the attention weights assigned by models trained in a fact verification task. Here we show that attention weights tend to be directed towards words and noun phrases that are more likely to be domain specific, and it considerably impacts performance in an out-of-domain setting.

Next, to mitigate this dependence on lexicalized information we devise some novel de-lexicalization methods that replace concepts with their respective semantic classes Panenghat et al. (2020); Suntwal et al. (2019a). While these techniques of delexicalization (or masking) have been used before (Zeman and Resnik, 2008), we have expanded it by incorporating semantic information (the assumption that meaning arises from a set of independent and discrete semantic units) (Peyrard, 2019). Since these techniques are general and compatible with most existing semantic representations, we believe they can be further extended onto datasets used for other NLI tasks. Thus, by enabling integration of these techniques into the training

pipeline, we hope to control lexicalization in the datasets which the neural network methods possibly depend upon.

In particular, we delexicalize two datasets used in Fact Verification, the FEVER and Fake News Challenge datasets. These datasets are delexicalized using several strategies which are based on human intuition and underlying linguistic semantics. For example, in one of these techniques, lexical tokens are replaced or masked with indicators corresponding to their named entity class (Grishman and Sundheim, 1996). Our work differs from early works on delexicalization (Zeman and Resnik, 2008) in that we earmark overlapping entities (between the hypothesis and premise) with a unique id. Further we also explore semantic lifting to WordNet synsets using Super Sense tags (Ciaramita and Johnson, 2003; Miller et al., 1990).

Further, we analyze and examine the effect of such delexicalization techniques on several state of the art methods in fact verification and confirm that these methods can still achieve comparative performance in domain. We also show that in an out-of-domain set up, the model trained on delexicalized data outperforms that of the state of the art model trained on lexicalized data. This empirically supports our hypothesis that delexicalization is a necessary process for meaningful machine learning. Thus we show that altering the fact verification datasets based on lexical importance is beneficial for organizing research and guiding future empirical work in this field.

### 1.3.2 Creating a neural network based method to control over delexicalization

In the first chapter we show that our data distillation methods encourage neural networks to look beyond lexical patterns and extract underlying semantic knowledge that can be transferred from one domain to another. However, a drawback to this approach is that we don't know ahead of time what the successful strategy for level of delexicalization is. While a little delexicalization helps reduce over-fitting, too much delexicalization might discard useful information.

As a solution to this we propose model distillation, a novel architecture based on the teacher-student paradigm Hinton et al. (2015). In this architecture a teacher and

#### 1.4. CREATING AN ARCHITECTURE TO AUTOMATICALLY ARRIVE AT THE RIGHT AMO

student neural network method compete with each other to arrive at the right level of abstraction needed for data distillation. Due to its ability to transfer learning across various domains, this approach aims to reduce the computational overhead and carbon footprint caused by having to train neural network based methods every time they need to make predictions on a new fact verification input. We show that that our method is not only efficient but also surpasses the current limit achieved by state of the art methods which do not use such abstraction techniques.

#### 1.4 Creating an architecture to automatically arrive at the right amount of delexicalization needed

While the aforementioned delexicalization techniques reduced over fitting, a key unresolved issue in this direction was *how much* delexicalization should be applied? We next show that combining these techniques with a model distillation strategy will arrive at the right level of delexicalization needed.

In particular, we propose a novel architecture called group learning architecture, inspired from the teacher-student architecture mentioned in section 1.3.3. Specifically, in our architecture we use multiple copies of the same model, but each having access to different perspectives of the same data. Each such data versions are delexicalized with a different level of granularity. Additionally one model has access to the original plain-text data. The *intuition* behind this approach is that the proposed architecture will enable various models to “pull” towards the original underlying semantics, which are partially obscured to them due their respective delexicalized versions of data. These models then will compete with each other, and also learn from each other through pair-wise consistency losses, to arrive at the right level of abstraction needed for data distillation.

Further we show that not only is our approach classifier agnostic, it also transfers learning across various domains (?). Further, this approach reduces the computational overhead (and, thus, carbon footprint) caused by having to train neural network based methods every time they need to make predictions on new fact verification domains or datasets.

## 1.5 Overview

The rest of this work is organized as follows. In Chapter 2 we outline the previous work relevant to our task. In Chapter 3 we detail the delexicalization techniques and show that it improves domain transferability. Then in Chapter 4 we present our work where we reduce over delexicalization using a student teacher architecture. In Chapter 5 we detail the approach that uses multiple models training simultaneously to arrive at an apt level of delexicalization needed for domain transfer. Finally in Chapter 6 we discuss the work as a whole as well as future directions.

## CHAPTER 2

### RELATED WORK

Recent years have witnessed a boom in research efforts in the NLP field that target machines' capacity to do deep language comprehension. This goes beyond what is clearly expressed in text, instead relying on reasoning and knowledge of the world. Many benchmark tasks and datasets have been developed to aid in the development and assessment of such natural language inference ability.

#### 2.1 Natural Language Inference

Originally the phrase natural language inference(NLI) was used to refer to statements which needed more contextualized or background knowledge for the machines to understand them. For example consider this example cited in the seminal (Minsky, 2000) work on common sense based inference: “Jack needed some money, so he went and shook his piggy bank. He was disappointed when it made no sound.” This is a statement that is easily understood by humans but not by machines. A human has the necessary background knowledge that piggy bank is a kind of money saving device, and it not having any coins (another form of money) meant that the owner had no savings, and hence is meant to get disappointed. However, it is difficult for machine to comprehend this entire meaning unless it has access to prior knowledge. This, notoriously difficult task for machines, is termed as natural language inference. i.e to have an ability to deeply understand natural language beyond what is explicitly expressed.

History of NLI is closely entangled with many benchmark datasets (and associated challenge tasks) that have been created to help develop and evaluate NLI algorithms and models. These datasets have drawn significant attention from the

research community and many learning and inference approaches have been developed.

### 2.1.1 Datasets

Creating benchmark datasets to evaluate machines' progress on language processing tasks and promote the development of novel approaches to the natural language inference task has been a long-running task. However, many early attempts were on gathering large-scale annotations to assist data-driven techniques for sub tasks such as part of speech tagging (Marcus et al., 1993), named entity recognition (Grishman and Sundheim, 1996), information extraction (Sundheim, 1993), co-reference resolution and relation extraction (Doddington et al., 2004). While these datasets were created primarily to investigate the data contained in linguistic knowledge and context, other datasets were created to focus on reading comprehension through tasks like question answering (Hirschman et al., 1999). To answer questions possible through shallow linguistic approaches, these early reading comprehension benchmarks frequently required only one sentence of context at a time.

In recent years, benchmark datasets (and related tasks) have shifted away from using language context and moved towards requiring a deeper knowledge to complete these tasks. For example consider the sentence from the Winograd schema challenge for evaluating progress in commonsense reasoning (Morgenstern and Ortiz, 2015): "The trophy would not fit in the brown suitcase because it was too big. What was too big?". Linguistic constraints alone will not be able to determine if this question relates to the trophy or the brown suitcase. Hence more recent datasets have been built on the belief that successfully solving these tasks will need machines to go beyond language context and rely on reasoning and information not directly conveyed in text. However, the scope of these benchmark datasets vary from the specific task of co-reference resolution to broader tasks like textual entailment. While some benchmarks are focused on a single task (e.g., coreference resolution task in the Winograd Schema Challenge (Morgenstern and Ortiz, 2015)), others, such as GLUE (Wang et al., 2018), are made up of a variety of tasks. While some benchmarks are

designed to address a single type of knowledge, e.g., commonsense social psychology in Event2Mind (Rashkin et al., 2018), some others like the CLOZE evaluation for deeper understanding of commonsense stories (Mostafazadeh et al., 2016) are designed to cover a number of forms of knowledge, which necessitates a wide range of common facts. Some datasets contain multiple-choice questions that require a binary decision or a candidate list selection, while others are more open-ended. Different benchmark features serve different objectives. Therefore determining what criteria to examine in order to establish a benchmark that can assist technological development and provide a quantitative consequence of research progress on deep reasoning abilities is essential. While a comprehensive review of all benchmark tasks that comprise NLI are beyond the scope of this work, in section 2.2 we will talk about benchmarks used for a specific sub task very relevant to the current work, Recognizing Textual Entailment (Dagan et al., 2013).

## 2.2 Recognizing Textual Entailment

Recognizing Textual Entailment (Dagan et al., 2013) (RTE), is the task of determining whether the meaning of one text fragment can be inferred (or entailed) from the meaning of the other, given two text fragments. It is also be described as the directed relationship between a text and a hypothesis in which the text entails the hypothesis, if a normal person would conclude that the hypothesis is true based on the text. In section 2.2.1 we will talk about a few RTE challenges and benchmark datasets which advanced the field by giving rise to certain prominent systems, which will then be described in section 2.2.2 .

### 2.2.1 Datasets

For many years now, the RTE problems have been the dominant reasoning task under the NLI spectrum<sup>1</sup>, inspired by difficulties faced in semantic processing for

---

<sup>1</sup>More recently, especially since the release of Stanford Natural Language Inference dataset (Bowman et al., 2015)), the phrase natural language inference is even interchangeably used to denote the task of recognizing textual entailment (RTE).

machine translation, question answering, and information extraction (Dagan et al., 2005). While the initial benchmarks for this task construed of recognizing only *entailment*, some later ones explicitly expected the prediction of *contradiction* also, the opposite label of entailment. Specifically, while in first three RTE challenges (Dagan et al., 2010; Haim et al., 2006; Giampiccolo et al., 2007) the systems were expected to predict whether the text entailed the hypothesis or not, in the fourth and fifth Challenges (Bentivogli et al., 2009; Giampiccolo et al., 2008), systems were also expected to identify contradiction connections between texts and hypotheses as part of a new three-way classification task. The major objective for the sixth and seventh Challenges (Bentivogli et al., 2011) were to find entailment given one hypothesis and a corpus with numerous possible entailing phrases. The eighth challenge (Dzikovska et al., 2013) focused on categorizing student responses in order to provide automated feedback in an educational context, which was a slightly different challenge.

The Conversational Entailment Dataset (Zhang and Chai, 2010) which was introduced after this, consisted of a short series of 875 binary entailment pairs. These consisted of premise and hypothesis pairs which were created from natural language conversation and hence required the solving of interpreting dialogue. Specifically, many examples required anticipating the beliefs, wishes and intentions of conversation participants that must be explained in a dialog setting. Sentences Involving Compositional Knowledge (SICK) was another benchmark dataset which had around 10,000 premise hypothesis pairs (Marelli et al., 2014). However, there were two subtasks, one to find sentence relatedness and the second one, the classic entailment task. The Stanford Natural Language Inference dataset, introduced in the year 2015, had 60,000 sentence pairs (Bowman et al., 2015). The specific task was similar to the fourth and fifth RTE challenges, (Bentivogli et al., 2009; Giampiccolo et al., 2008), i.e to make a 3-way prediction of these pairs into classes of *entailment*, *neutral* and *contradiction*. Further to these 3 classes, the SNLI dataset also included five labels to denote the agreement between the annotators. Later, this benchmark was expanded to a new dataset named MultiNLI (Williams et al., 2017), which is in the same structure but includes phrases of many genres including telephone conver-

sations. MultiNLI is available as part of the broader GLUE benchmark (Wang et al., 2018). SciTail is another entailment dataset that consists of about 27,000 pairs of premise hypotheses tailored to a 2way entailment problem from science questions (Khot et al., 2018). This is mainly science-based, and may demand domain-specific expertise, unless alternative benchmarks are used.

### 2.2.2 Methods for Recognizing Textual Entailment

A number of techniques have been developed to accomplish the benchmark tasks mentioned in Section 2.2.1. These span from the earlier symbolic and statistic techniques to the use of more recent deep learning and neural networks. In this section, we will provide a quick review of the early logic based and statistical methods, followed by a more thorough explanation of typical neural techniques, which are the current state of the art on all of the benchmarks.

#### **Logic Based Methods**

While logic based methods for human reasoning have continued through history from the times of Aristotle, it wasn't until (Peirce, 1883) introduced logical abduction that reasonable inference was used for language issues. Logical abduction is the process of reaching a conclusion based on a small number of observations and a small number of assumptions. This is similar to the plausible inference for language problems as defined by (Davis and Marcus, 2015). Meanwhile, other logic theories found use in AI, such as McCarthy's early work on commonsense for machines (McCarthy et al., 1960), and linguistics, such as Lakoff's theory of natural logic toward a semantic description of language (Lakoff, 1970). Invention of fuzzy logic (Zadeh, 1988), which maps linguistic description of numeric variables to probability distributions over numeric variables, was another significant progress towards managing inaccuracies in human language. Moore (Moore, 1982) and Nunberg (Nunberg, 1987) further motivated the importance of reasoning in AI and the role of logic in common sense reasoning. Symbolic approaches to the representation of knowledge and the semantic processing of language were also prevalent throughout the early

1990s (Birnbaum, 1991). As a consequence, these techniques, especially symbolic approaches, were utilized extensively for early NLI challenges and later for RTE challenges. For example in (Raina et al., 2005) sentences are parsed in a logical manner and then abduction is performed over them using learned assumptions and probabilities to establish if a very similar set of assumptions may be utilized to demonstrate that a statement entails another. In another similar work Giampiccolo (Giampiccolo et al., 2008) supplemented information in the hypothesis sentence with outside semantic knowledge from resources such as Wikipedia, WordNet (Miller, 1995), and VerbOcean (Chklovski and Pantel, 2004), and then attempted to map this to terms in the premise phrase using manually created logic rules. In a subsequent work (MacCartney and Manning, 2007) turned natural logic into a formalism for NLI, parsing the premise and hypothesis into natural logic form and utilizing a decision tree to compare features and make a decision. On the third RTE Challenge dataset (Giampiccolo et al., 2007), they demonstrate that combining this technique with an existing statistical RTE model outperforms the state-of-the-art performance. Another similar technique that yields high performance develops a comprehensive collection of hand produced logic and commonsense rules to apply with supplied mappings from plain language in benchmark data to logical forms (Gordon, 2016). While it has been demonstrated to be extremely efficient in some tasks to manually draw up logical rules and mappings from a language to a logical form, this cannot be scaled for more modern large datasets, in which there is more variation in required knowledge and language and much higher semantic phenomena. To mitigate this (Kamath and Das, 2018) developed certain techniques, known as semantic parsing, which automatically map natural language text to a logical form, in order to allow more practical, scalable, logic based reasoning.

### **Statistical Methods**

From mid-1990s till early 2010 many statistical techniques dominated the NLP field. These earlier statistical techniques relied on engineered features to train various types of statistical models based on training data for a number of early NLP bench-

marks. For example, early methods frequently coupled different word matching and other lexical features with traditional statistical models such as decision trees (Ng et al., 2000) or with hand crafted deterministic rules (Charniak et al., 2000).

Similar techniques were used in several of the RTE benchmarks, beginning with the RTE Challenges (Dagan et al., 2005). Lexical characteristics based on bag-of-words and word matching, for example, were frequently utilized in the first two RTE challenges (Dagan et al., 2005; Haim et al., 2006), but yielded only marginally better outcomes than random guessing. Additionally, to generate predictions some other systems have employed more linguistic features, such as, hidden correlation biases in benchmark data, synonym, antonym, and hypernym relationships derived from training data (Lai and Hockenmaier, 2014), semantic dependency and paraphrases (Hickl et al., 2006) etc.,. Another category of solutions use external data sources and the Internet to augment training data characteristics. For instance, a naive Bayes classification system utilizing features from the co-occurrences of word on a web search engine (Glickman, 2006) was the best system in the initial RTE Challenge (Dagan et al., 2005). In order to assess the statistical measure of entailment between the given phrases, the top system for the seventh RTE Challenge (Tsuchida and Ishikawa, 2011) also utilised some external knowledge resources such as WordNet (Miller, 1995) and CatVar (Habash and Dorr, 2003) along with the acronyms produced from the training data. Even though the use of some external knowledge bases have an advantage over models that simply employ language characteristics derived from the data, they have not been competitive enough in the recent benchmarks of enormous data size. Nevertheless, as demonstrated by (Zhang et al., 2017), they continue to be used as useful baselines for new benchmarks.

Aided by many factors including the rising amount of data for current benchmarks, the Internet, acceleration in software and hardware discoveries etc., larger and deeper neural networks are being invented and trained day by day. As of writing this work, neural network based solutions are the state of the art in most of the NLI benchmarks discussed so far. Next, we will elaborate on some common techniques and architectures that contributed to this journey.

### Neural Methods

Neural methods owe their origins to past statistical approaches used for predictions. However, instead of manually describing all the features like the past statistical approaches did, it utilizes different types of architectures to find valuable features in the data. One such feature recognition method, a crucial method that accelerated the rise of neural networks, was the invention of the distributional representation of words like Word2vec (Mikolov et al., 2017) and Glove (Pennington et al., 2014). The word vectors (also known as embeddings) representing features are frequently learned (and in some cases updated) from large text corpora using neural networks. However, one shortcoming of these early approaches were that the embedding vector of the target word is always the same, regardless of the context in which it occurs. As a result, these embeddings were incapable of representing distinct word meanings in different contexts, despite the fact that this is a common occurrence in language. Recent works, such as Embeddings from Language Models (ELMO)(Peters et al., 2018) and Bidirectional Encoder Representations (BERT) (Devlin et al., 2018) from Transformers (Vaswani et al., 2017), have created contextual word representation models to solve this challenge. These approaches assign various embedding vectors to words depending on their context.

These pre-trained word representations can be fine-tuned for downstream tasks or utilized as features. For example, the Generative Pre-trained Transformer (GPT) (Radford et al., 2018) and BERT (Devlin et al., 2018), introduce few task-specific parameters and may be fine-tuned on downstream tasks with modified final layers and loss functions. Further, task-specific network architectures are built for various downstream applications on top of these word embedding layers. These task-specific architectures used to handle specific tasks frequently include recurrent neural networks (RNNs) such as LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014) convolutional neural networks (CNNs), or more recently, transformers (Vaswani et al., 2017).

These networks' output layers are chosen based on the task specification. A linear layer and softmax, for example, are frequently used for classification, whereas a

language decoder is frequently used for language creation. RNN-based architectures are extensively and frequently used in both baseline (Bowman et al., 2015; Rashkin et al., 2018) and state-of-the-art techniques (Kim et al., 2019; Chen et al., 2018; Henaff et al., 2016) because of the sequential structure of language. Neural models benefit from approaches like attention mechanisms and memory augmentation when they have diverse topologies. Memory-augmented networks, such as memory networks (Weston et al., 2014) and recurrent entity networks (Henaff et al., 2016), have been proven to be successful for tasks that require reasoning based on numerous supporting facts (e.g., bAbI (Weston et al., 2015)).

Since its first usage in neural machine translation (Bahdanau et al., 2014), attention has been frequently employed in NLP tasks. Next we will elaborate on a few neural models for the aforementioned benchmarks in question, with a particular focus on those based on the attention mechanism, since they comprise most of the past and current state-of-the-art models on these benchmarks.

### **Attention**

Attention is a mechanism which was first used to capture the alignment between an input (encoder) and an output (decoder). In this mechanism, the decoder chooses which portions of the original phrase to pay attention to. Thus by allowing the decoder to have an attention mechanism the encoder is alleviated from the burden of needing to encode every information in the source phrase into a fixed length vector. With this novel technique, information may be dispersed across the sequence of annotations, and the decoder can extract it selectively as needed.

There are numerous advantages to modeling attention. As mentioned above, it enables the decoder to go straight to specific sections of the input and focus on them. It solves the vanishing gradient problem by allowing for states further in the input sequence to be accounted for. Another advantage is that the model's learned attention allocation automatically aligns inputs and outputs, allowing some comprehension of their relationships.

Adding an attention mechanism to RNNs, LSTMs, CNNs, and other deep learn-

ing models has been demonstrated to enhance performance on a variety of tasks when compared to their vanilla counterparts (Kim et al., 2019). It works especially well for tasks that need input and output alignment, such as different textual entailment tasks that require context and hypothesis modeling. One of the best performing systems in the SNLI challenge, for example, employs a densely connected RNN while concatenating features from an attention mechanism to recurrent features in the network (Kim et al., 2019). The attention weights that arise from this alignment, as explained by (Kim et al., 2019), aid the system in making correct entailment and contradiction judgements for very similar pairs of phrases. One such example they give is when the hypothesis statement “Several individuals in front of a gray building” is compared to the context sentence “Several guys in front of a white building.”

In contrast to the aforementioned architectures which add attention mechanisms to neural architectures like CNN, RNN etc., is the recently proposed Transformer architecture which is entirely composed of attention mechanisms (Vaswani et al., 2017). The self-attention layer in both the encoder and the decoder is the fundamental distinction in Transformer networks. Self-attention permits it to attend to all positions in an input sequence to better encode the word. It provides a mechanism for possibly capturing long-term word dependencies such as syntactic, semantic, and coreference relations. Furthermore, rather than performing a single attention function, the transformer performs multi-head attention. i.e., it applies the attention function multiple times with different linear projections. This allows the model to jointly capture different attentions from different subspaces, e.g., jointly attend to information that may indicate both coreference and syntactic relation. Another significant advantage of the transformer is its capacity to support parallel processing. Because of their sequential nature, sequence models such as RNN and LSTM are challenging to parallelize. The transformer model meanwhile, by using attention to capture global relationships between inputs and outputs, maximizes the number of parallelizable computations. Further, Transformers have lately been employed in pre-trained contextual models like (Radford et al., 2018) and BERT

(Devlin et al., 2018) to attain state-of-the-art performance on several of the RTE benchmarks mentioned earlier.

### 2.3 Fact Verification

The task of fact verification is determining if assertions made in written or spoken language are correct. The fact checker, typically a journalist, must analyze prior remarks, debates, laws, and published numbers or known facts, and utilize them, together with logic, to make a conclusion. However, this is made difficult in recent times due to the Internet’s ability to swiftly transmit information. This offers a great opportunity for individuals and organizations to disseminate and consume enormous volumes of material on nearly any subject area. This increased ability to reach audiences is now being used to spread both true and false information. It has become a major concern because recent studies have shown that false information reaches a larger audience and has influenced public decisions like voting (Vosoughi et al., 2018). Because of the growing need for fact checking, considerable progress has been made in developing tools and systems to automate the job, or sections of it. As a result, a wide range of methods have been developed, each customized to certain datasets, domains, or subtasks (Babakar and Moy, 2016). In this section we will look at some datasets and methods which have advanced the field of automated fact checking from the perspective of natural language processing (NLP).

#### 2.3.1 Datasets

In this section first we will look at various datasets used in fact verification chronologically. Then we will look at these datasets from various perspectives based on the style and content of the inputs and outputs to the automated fact-checking system.

Fact verification datasets started appearing in public around mid 2010s, and since then there have been only a limited number of published datasets. Each of these had different features and characteristics for claims and evidences as seen in the previous sections. The earliest known instance of a fact verification dataset was

released by (Vlachos and Riedel, 2014) which had 221 labeled claims in the political domain that had been fact checked by organizations like Politifact and Channel4<sup>2</sup>. The claim’s originator was included alongside hyperlinks to the sources which could be utilized by a human fact checker as evidence for each labeled claim. These hyperlinks further led to various sources, including but not limited to, pdf reports and documents from the National Archives, statistical tables, Excel spreadsheets etc.,. The shortcoming of this dataset was that, while the provided claims were readily verifiable by human fact checkers using the hyperlinks provided, the lack of a systematic structure in the type of evidence document made it unusable for a machine learning based fact verification.

In the year 2016 (Ferreira and Vlachos, 2016) created a rumor debunking dataset based on data from the Emergent project (Silverman, 2015). For each of the 300 claims, 2,595 news items were gathered, and their stances were categorized as *for*, *against*, or *observing* the claim. In 2016 itself, as part of the HeroX Fast and Furious Fact Checking Challenge (Francis, 2016) published 90 (41 training and 49 test) labeled statements. Because part of the challenge required the identification of suitable source material, no evidence for these assertions was supplied, necessitating case-by-case manual review. The dataset’s size, like that of (Vlachos and Riedel, 2014), prohibited it from being used to train a machine learning-based fact-checking system. However, the wide range of claims in this dataset exposed a variety of sorts of disinformation, making it easier to identify the needs for fact-checking systems.

Inspired by this, in 2017, (Wang, 2017) released a similar dataset with 12 thousand labeled claims from Politifact, an order of magnitude higher than (Vlachos and Riedel, 2014). Further, this dataset provided meta-data such as the speaker affiliation and the context in which the claim appears in (e.g. speech, tweet, op-ed piece) to be used as evidence. Due to its size this dataset was easily used to train and evaluate a machine learning-based fact-checking system. However, the dataset’s use was restricted due to the lack of machine-readable evidence beyond source metadata, which means that systems were limited to using techniques like text classification

---

<sup>2</sup><http://channel4.com/news/factcheck>

or speaker profiling. Similarly (Rashkin et al., 2017) not only compiled Politifact claims without using meta-data but also released a dataset comprising 74K news stories collected from websites categorized as *hoax*, *satire*, *propaganda*, and *trusted news*. According to a US News & World report <sup>3</sup>, the challenge of determining whether a news story is real or fraudulent is represented as predicting if an article originated from one of the websites regarded to be “fake news.” However, since it did not provide explicit evidence sentences, machine learning systems were limited by the learning they can acquire from linguistic features of the provided claims only. In late 2017, for the Fake News Challenge, (Pomerleau and Rao, 2017) expanded the Emergent dataset (Ferreira and Vlachos, 2016) to include roughly 50K headline and body pairs generated from the initial 300 claims and 2,595 articles.

In 2018, (Thorne et al., 2018b) published a fact-checking dataset including 185K claims regarding entity and concept characteristics that had to be validated using Wikipedia articles. While this significantly reduced the pool of evidence compared to HeroX (Francis, 2016), it did provide for a tough evidence selection sub task that could be annotated manually on a wide scale in the form of sentences chosen as evidence. These machine readable fact checks, when combined with the labels on the claims, allowed machine learning-based fact checking models to be trained.

Next we will look at the same datasets based on how they handle inputs and outputs. Looking at the fact verification datasets from the type of claims, evidences and outputs is important, since it in-turn influences the sorts of claim and evidence present in these datasets.

### Types of claims

Representing claims as subject-predicate-object triples, such as (London, capital of, UK), are a common input to fact-checking techniques and are popular across many research communities. The appeal of triples as input derives from the fact that they make fact checking with (semi-)structured knowledge stores like Freebase

---

<sup>3</sup>[www.usnews.com/news/national-news/articles/2016-11-14/avoid-these-fake-news-sites-at-all-costs](http://www.usnews.com/news/national-news/articles/2016-11-14/avoid-these-fake-news-sites-at-all-costs)

easier (Bollacker et al., 2008). However, the short coming of these methods is that using triples as input implicitly demand a high degree of processing to transform text, voice, or other kinds of claims into triples, a complicated job that falls within the wide notion of natural language comprehension.

Textual claims are a second form of input that is frequently examined in automated fact checking. These are usually brief phrases created from larger paragraphs, which is a frequent method used by human fact checkers on specialized websites like PolitiFact<sup>4</sup> and Full Fact<sup>5</sup> to include just the context relevant to the assertion from the original material. Further, in the context of the HeroX fact-checking challenge, a taxonomy of textual assertions was developed. According to this, claims were classified into four types: numerical claims, entity and event properties such as professional qualifications and event participants, position statements such as whether a political entity supported a certain policy and quote verification assessing whether the claim states precisely the source of a quote, its content, and the event at which it supposedly occurred.

Finally, some other techniques take the full document as input. However, the short coming of these methods is that they must first identify the claims, after which they must be fact-checked. This complicates the job since claims must be extracted in the form of triples using relation extraction (Vlachos and Riedel, 2015) or through a supervised sentence-level classification (Hassan et al., 2015).

### **Types of Evidences**

The model and the sorts of outputs that the fact-checking system can provide are influenced by the type of evidence used for fact-checking. The fact-checking system's ability to create a label or a justification, for example, is mainly determined by the information accessible to it. The first type are the task formulations that predict the validity of a claim without using any evidence other than the claim itself (Rashkin et al., 2017). In these cases, the predicted veracity is linked to surface-level language

---

<sup>4</sup><http://www.politifact.com/>

<sup>5</sup><http://www.fullfact.org/>

characteristics in the assertions. The downside of these methods is that predictions are based on surface patterns of how the claim is worded rather than evaluating the current condition of the world. Further, (Wang, 2017) incorporates additional metadata such as the claim’s originator, speaker profile, and the media source in which the claim is presented. While the extra background does not provide evidence to support the claim, it can be utilized as a prior to enhance classification accuracy and as part of the reasoning for a judgement.

Another prevalent methodology is utilizing knowledge graphs for fact-checking. Knowledge graphs are a comprehensive collection of organized canonical knowledge about the world recorded in a machine-readable manner. Using this sort of data, we notice two types of formulations. The first method is to locate/retrieve the element in the knowledge graph that has the data that supports or refutes the assertion in issue. (Vlachos and Riedel, 2015), for example, uses tiny knowledge networks to identify subject-predicate-object triples in order to fact-check numerical assertions. Following the discovery of the relevant triple, a truth label is calculated using a rule-based method that takes into account the difference between the claimed values and the retrieved values from the graph. The main drawback of utilizing knowledge graphs as evidence in this way is the assumption that it encompasses all the facts relevant to the claim. However, without first understanding the claim, it is impossible to collect and store every possible fact in the graph. The second method to utilize a knowledge graph as evidence is to use its topology to forecast the likelihood of a claim (represented as an edge in the graph) being true (Ciampaglia et al., 2015). While the topology of a network might indicate the plausibility of a fact, the fact that it is improbable to happen does not undermine its accuracy. Furthermore, as seen from the recent covid vaccine related misinformation spread, improbable but believable claims are more likely to become *viral*, necessitating further fact-checking by fact-checkers.

Other sources, text from which can be used for the verification of claims, include encyclopedia entries, policy papers, verifiable news, and scientific publications. For example, news article headlines (single sentences) were used by (Ferreira and Vla-

chos, 2016) to predict whether an article is for, against, or discusses a claim. The Fake News Challenge (Pomerleau and Rao, 2017) also used this fact-checking procedure. However, unlike (Ferreira and Vlachos, 2016), it used full documents as evidence allowing the combination of multiple sentences to be used as evidences.

Another category of methods for fact verification includes retrieval of evidence information *after* the claim is provided. For example, the Fact Extraction and VERification (FEVER) task (Thorne et al., 2018a) required combining data from different documents and sentences. However, unlike the aforementioned works that provided evidence text, in FEVER task, the evidence is not provided; instead, it must be retrieved from Wikipedia, a vast collection of encyclopedic entries. The WSDM cup’s triple scoring assignment (Bast et al., 2017) took this methodology even a step further, by requiring participants to evaluate knowledge graph triples based on both Wikipedia and a piece of the web-scale ClueWeb dataset (Gabrilovich et al., 2013).

Another source of text-based evidence are repositories of previously fact-checked statements (Hassan et al., 2017). Systems employing such evidences will search for a claim in the repository similar to a new claim and return the truth label, if one is found. However, the short coming of this methodology is that it restricts fact-checking only to claims that are similar to those already present in the repository.

Another alternative to searching for knowledge in texts or databases is aggregating information on the distribution of posts on social networks to provide insights on the authenticity of the content (Derczynski and Bontcheva, 2014; Gorrell et al., 2019). This type of crowd behavior can give important information, especially when textual sources or organized knowledge bases are absent. This evaluation of macro level behaviors of users’ interactions with and dissemination of material to forecast whether the claims in the content are accurate or untrue is known as rumor veracity prediction.

However, despite its apparent relevance, automated techniques seldom address the reliability of the sources utilized as evidence, i.e. the verification component of fact checking. The sources of evidence are frequently assumed to be supplied, such as

Wikipedia or Freebase, because this makes creation and evaluation easier, especially when many models are being examined. Nevertheless, fact-checking methods that depend on Twitter metadata frequently evaluate credibility indications for the tweets used as evidence. However, we will not be discussing this trustworthiness assessment because it is often considered as a separate task and is hence beyond the scope of this work.

### Outputs of a Fact Verification Task

Next we will talk about the outcomes of such a fact verification task, which also vary from datasets to datasets and from specific tasks to tasks. For example, the simplest form of fact verification is labeling a given claim as true or false (Nakashole and Mitchell, 2014). However, in natural language, truth is never a clear binary classification and in fact varies across a scale. For example, fact checking agencies such as Politifact <sup>6</sup> model the degree of truthfulness on a multi-point scale (ranging from true, mostly-true, half-true, etc.). Such variations exist also in the world of fact verification using natural language processing. For example, (Vlachos and Riedel, 2014) suggested modeling this degree of truthfulness as an ordinal classification task. However, (Wang, 2017) and (Rashkin et al., 2017) model the expected output as multiclass labels following the definitions by the journalists over a multi-point scale but ignoring the ordering among them. The WSDM cup’s triple scoring job (Bast et al., 2017) expected triples to be scored within a numerical range indicating how likely they are to be true as output. In their 2016 task/dataset Emergent (Ferreira and Vlachos, 2016) expected the output as whether a claim is supported, refuted, or just reported by a news story headline. (Pomerleau and Rao, 2017) introduced an additional label for articles that are unrelated to the claim, and they take the whole article into account rather than just the title. Thus they had a total of four labels for the claim to be classified into: *supports*, *refutes*, *discuss* and *unrelated*. The expected output from FEVER (Thorne et al., 2018a) task was two-fold: First a three-way classification label indicating whether a claim is *supported* or *refuted*

---

<sup>6</sup><http://www.politifact.com/>

by Wikipedia, or whether there is insufficient information (labeled *nei*- not enough information) in the Wikipedia for it. Second, the systems were expected to return the specific sentences that were used to arrive at the verdict of, *supported* or *refuted*.

### 2.3.2 Methods

The bulk of automated fact-checking systems are supervised, i.e., they develop a text classifier from labeled data supplied at training. (Vlachos and Riedel, 2015) first proposed fact-checking with supervised models, based on existing statements annotated with judgements by fact-checking organizations and journalists. This method was later used by (Wang, 2017) and (Rashkin et al., 2017) to classify the authenticity of false news articles. The primary drawback of text categorization methods is that fact-checking a claim needs extra world information, which is often not supplied with the claim.

(Ciampaglia et al., 2015) models the fact verification task as a path ranking problem and uses network analysis to predict if an unobserved triple will emerge in a graph. The cost of travelling a path between the two entities under transitive closure, weighted by the degree of connectedness of the path, determines the truth verdict. (Nakashole and Mitchell, 2014) integrate linguistic information on the subjectivity of the language employed with the co-occurrence of a triple with other triples from the same topic, as a type of collective classification task. They show that language can help determine if a phrase is true or untrue. However, because of the complexity of natural language, credible-sounding sentences can also be false. Hence fact verification based on claims alone has not been employed in many systems. Instead it was used in the related problem of finding fact-check worthy statements, text classification on claims alone has been employed (Hassan et al., 2017).

Fact checking was modeled by (Ferreira and Vlachos, 2016) as a type of Recognizing Textual Entailment (RTE) problem. The problem then was about predicting whether a premise, typically (part of) a news article, is for, against, or observes a given claim. The same concept was used by many of the participants and winners

(Riedel et al., 2017; Hanselowski et al., 2017) of the Fake News Challenge (Pomerleau and Rao, 2017) also.

The RTE-based models have an inherent assumption that textual evidence to verify a claim will be provided. As a result, they are inapplicable in cases where this is not provided (e.g., in the HeroX challenge (Francis, 2016)). They are also inapplicable in cases where the textual resource is quite large, such as Wikipedia (as in FEVER) (Thorne et al., 2018a). The authors of FEVER task in turn developed a pipe-lined approach to solve this, in which the RTE component is preceded by a document retrieval and a sentence selection component. The work of (Hua and Wang, 2017) focus solely on retrieving sentence-level evidence from related documents for a given claim .

To identify surface patterns in text that describe relations between two entities in a knowledge graph, (Vlachos and Riedel, 2015) and (Thorne and Vlachos, 2017) utilize distantly supervised relation extraction (Mintz et al., 2009). Matching a claim with existing, previously fact-checked claims is another frequently used model by fact-checking organizations in their process. As suggested by (Vlachos and Riedel, 2014) and implemented in ClaimBuster (Hassan et al., 2017) the task now simplifies into to sentence-level textual similarity. The same method was also used in the TruthTeller application by The Washington Post <sup>7</sup>, and in one of the two modes of Full Fact’s Live platform <sup>8</sup>. However, the short coming of this method is that it can only be used to fact-check claims that are previously repeated or paraphrased.

(Gottipati et al., 2013) showed that speaker profiling is useful for fact-verification. Inspired by this (Long, 2017) extended the system by (Wang, 2017) extended profiling of the originators of the claims. The credit history of the originator, a measure reflecting how often the originator’s claims are labeled as fraudulent, is the most significant factor in this model. However, this functionality would be inaccessible in the event of previously unannotated sources with no known history. Additionally, the heavy dependence on credit history has some significant ethical concerns that

---

<sup>7</sup><https://www.washingtonpost.com/news/ask-the-post/wp/2013/01/29/introducing-truth-teller-political-fact-checking/>

<sup>8</sup><https://fullfact.org/blog/2017/jun/automated-fact-checking-full-fact/>

must be carefully examined.

## 2.4 Domain Adaptation

Traditional supervised learning methods are based on the assumption that the training and test data are chosen from the same distribution. i.e., training and test data are assumed to be independently and identically (i.i.d.) sampled from the same underlying distribution. In practice, this assumption does not hold, which translates into a drop in performance when the model trained on a source domain is tested on a different but related target domain. This mismatch of distributions between source and target domain is typically referred to as a domain shift (Daume III and Marcu, 2006). Consider the following scenario: a significant amount of manually labeled data is available in one domain, but the objective is to generate predictions on instances from a different domain with little or no labeled data. Because the examples in the two domains are taken from different distributions, classical supervised learning cannot achieve good performance on the new domain. For example, in the part-of-speech tagging task, the word *monitor* is more likely to be a verb in the financial domain, but it is more likely to be a noun in the corpus of computer hardware. It is to solve this issue that domain adaptation methods were conceived i.e., these methods are meant to overcome the distribution gap between training and test data. In this section, we investigate the motives, problems, and several domain adaptation methods in the field of natural language processing (NLP).

Also at this point we would like to elaborate a bit on some related terminology. Domain adaptation is considered closely related to transfer learning in literature (Pan and Yang, 2009). Transfer learning is a broad name for a group of machine learning issues involving several tasks or domains. There isn't currently a common definition of transfer learning in the literature and in the past it has been used interchangeably with domain adaptation. More recently domain adaptation is considered to be a particular case of transfer learning, namely transductive transfer learning (Ruder, 2019).

The task of domain adaptation is important because it is expensive in many

applications to collect labeled data in a new target domain, although there might be a significant amount of data accessible in certain source domains. For example, for well known news datasets, such as Penn Tree Bank <sup>9</sup>, and CoNLL Named Entity Recognition corpus <sup>10</sup>, manual annotations for part of speech tagging, syntactic parsing, named entity recognition, relation and event detection are readily available. However, training data for other genres and domains, such as emails, web forums, and biological articles, is rarely accessible. For example, gathering labeled data in the new domain (e.g., emails) is a difficult task due to the lack of subject matter experts/human annotators (or it is an expensive undertaking). This is where the utility of domain adaptation algorithms, that transfer knowledge from labeled data in the news domain to the new email domain, becomes important. Such approaches can reduce the need for costly manual annotations in the new domain while maintaining high performance in the old domain.

A part of domain adaptation research has been on supervised domain adaptation (Daumé III, 2009; Plank, 2011). A small quantity of labeled target domain data is provided in such a classic supervised domain adaptation setup, coupled with a larger amount of labeled source domain data. In light of limited target domain data, the goal is to adapt from the source to the specific target domain. As mentioned above annotation, is a time-consuming and costly manual endeavor. While annotation immediately addresses the absence of labeled data, it is difficult to expand to new application targets. Domain adaptation techniques attempt to move models' ability from typical interpolation of comparable cases to models that extrapolate to examples beyond the original training distribution Ruder (2019). Domain adaptation, as opposed to sequential transfer learning, aims to acquire representations that are helpful for a specific target domain rather than being useful in general. While there exists some supervised learning methods domain adaptation is generally studied in the unsupervised setting. Here a sufficient number of labelled examples in the source domain and only unlabelled examples in the target domain are assumed to be

---

<sup>9</sup><https://catalog.ldc.upenn.edu/LDC99T42>

<sup>10</sup><https://www.clips.uantwerpen.be/conll2003/ner/>

available, whereas sequential transfer learning requires access to labelled instances of the target task. A subclass of techniques investigates the supervised situation, which involves a limited number of labelled target instances.

Later in this section we will talk about three significant (and orthogonal) elements have been found to have an impact on domain adaption performance. The first element is model performance on the source task, which benefits from recent advances in neural models. The second factor is the variation in labeling functions between domains, which is due to the dataset’s inherent nature and should be small in practice. The third component is a measure of data distribution divergence; if the data distributions in the source and target domains are comparable, a model trained on the source domain should perform well on the target domain.

Having the source and target domains known a-priori is the typical setup for a classic domain adaptation task. However, domain adaptation is most commonly used in a few other related problems also: *cross-lingual learning*, *domain generalization/robustness*, and *out-of-distribution generalization*. In *cross-lingual learning* the model is trained on one language and is expected to predict in another language. The feature space in cross-lingual learning changes dramatically because alphabets, vocabularies, and word order might differ.

Rather than adapting to a specific target, *domain generalization/robustness* creates a single system that is resilient across multiple known target domains. The SANCL shared task (Petrov and McDonald, 2012) is one example, in which participants were required to create a single system that can robustly analyze reviews, weblogs, replies, emails, and newsgroups. The challenge in this arrangement boils down to creating a more robust system for specified objectives. It is possible to think of it as optimizing for both in-domain and out-of-domain(s) accuracy.

Domain adaptation is inextricably linked to a larger unresolved issue in machine learning. Intelligent systems should be able to adapt and manage any test distribution without ever seeing any data from it. This is the larger requirement for *out-of-distribution generalization* (Bengio et al., 2021), as well as a more difficult setup aimed at dealing with unknown domains (Volpi et al., 2018; Krueger et al.,

2021)).

Next we will talk about various methods used for domain adaptation specifically. Methods for solving domain adaptation range from classical methods (Jiang, 2008) to neural network based methods. These methods can be classified based on whether they modify the model, data or both.

#### 2.4.1 Methods that modify models

The methods that modify the model can further be broadly classified based on how they redesign various parts of the model to solve the task. For example, some models modify the features space, some the loss function, regularization, structure of the model etc.

##### **Methods that modify feature spaces**

Within feature-centric approaches, there are two distinct areas of research: feature augmentation and feature generalization. To create an aligned feature space, the former employs pivots (common shared features). Auto encoders are used to find latent representations that are more transferable across domains in the latter.

Structural correspondence learning (Blitzer et al., 2006) and spectral feature alignment (Pan et al., 2010) are two widely used techniques for methods utilizing feature augmentation or pivot-based techniques. Using unlabeled data from both domains, they both seek to discover characteristics that are similar across domains. The specifics of the process for constructing the shared area differ between the two approaches. For example, auxiliary functions (Ando et al., 2005) are used in structural correspondence learning, whereas a graph-based spectral learning approach is used in spectral feature alignment. Another seminal supervised domain adaptation technique that uses feature augmentation is EasyAdapt (Daumé III, 2009). It is based on the concept of creating domain-specific and domain-general features.

While structural correspondence learning was one of the early pioneering works in domain adaptation, it was dormant for a while, until recently when it was brought back to be used with neural networks (Ziser and Reichart, 2016, 2018b,a, 2019).

Specifically, (Ziser and Reichart, 2016) proposes using an auto-encoder structural correspondence learning model to combine the capabilities of pivot-based techniques with auto-encoder neural networks. Specifically, to map non-pivots to pivots, auto-encoders are used to learn latent representations, which are subsequently utilized to supplement the training data. However, the disadvantage of this method is that the text output vector representations are unique and context-independent. A pivot-based language modeling technique has been developed to tackle this issue (Ziser and Reichart, 2018a,b). This effectively integrates structural correspondence learning with a neural language model based on long short-term memory (Hochreiter and Schmidhuber, 1997) networks that predict the existence of pivots and non-pivots, resulting in structure-aware representations. However, the drawback of this method is the huge number of pivots required by the pivot-based language modeling method. To address this problem, a method of using task refinement learning technique based on pivot-based language modeling has been proposed (Ziser and Reichart, 2019). The method is an iterative training procedure in which the network is trained with a growing number of pivots. This showed improvements in accuracy and stability across a variety of hyperparameter selection options. Further, more recently there are hybrid methods proposed which add contextual embeddings to pivots (Ben-David et al., 2020). Also, the aforementioned techniques have another drawback of requiring two separate steps: one for representation learning and the other for task learning. To address this problem, current research suggests that the two tasks (pivot prediction and the actual task) be trained together and that pivots be learned automatically by attention, similar to work on automated non-pivot detection (Miller, 2019; Li et al., 2017, 2018). Note that, despite their efficacy, these methods have been used only for a few applications, including cross-lingual domain adaptation and sentiment classification (Ziser and Reichart, 2018b).

Feature generalization is the second of the common feature-centric approaches. Most of the methods that implement feature generalization use autoencoders. Autoencoders are neural networks that use an input reconstruction loss to learn latent representations from raw data in an unsupervised manner. The first study in this

line was by (Glorot et al., 2011), who developed the stacked denoising autoencoder for domain adaptation, which was in-turn motivated by denoising autoencoders (Vincent et al., 2008). In essence, a stacked denoising autoencoder learns a robust and unified feature representation for all domains by stacking many layers and intentionally corrupting the inputs with Gaussian noise, which the decoder must then reconstruct.

Initial architectures of stacked denoising autoencoders had problems with speed and scalability when dealing with high-dimensional data. To overcome these drawbacks, a more efficient marginalized stacked denoising autoencoder (MSDA) was suggested (Chen et al., 2012). Further, (Yang and Eisenstein, 2014) improved the regularization of MSDAs with marginalized structured dropout. (Clinchant et al., 2016) then improved the regularization of MSDAs using insights from domain adversarial training of neural networks (Ganin and Lempitsky, 2015). Despite its success, the primary drawback of autoencoder methods is that the induced representations do not make use of any linguistic information.

### **Loss based methods**

Loss-centric methods for domain adaptation can be further classified into two. The first approach uses domain adversarial networks and the second one uses level re-weighting methods. In this section we will briefly describe both these approaches.

Domain Adversaries based Methods: As per the theory of domain adaptation (Ben-David et al., 2010) cross-domain generalization may be accomplished using feature representations for which the origin (domain) of the input sample cannot be determined. Inspired by this (Goodfellow et al., 2014) proposed the Generative Adversarial Networks (GANs) which aims to minimize the discrepancies between training and synthetically generated data distributions. Further (Ganin and Lempitsky, 2015) used GANs for domain adaptation where they propose domain adversaries which aims at learning latent feature representations that serve at reducing the discrepancy between the source and target distributions.

Another seminal approach in the field of domain adaptation, called domain-

adversarial neural networks (DANNs) (Ganin et al., 2016), are inspired from the domain adversarial based methods mentioned above. The goal here is to find a task-specific accurate predictor while aiming to maximize the confusion of an auxiliary domain classifier in differentiating characteristics from the source and target domains. DANNs use a loss function via a gradient reversal layer to learn domain-invariant feature representations, ensuring that feature distributions in the source and target domains are made similar.

While the scalability and generality of this method are its strengths, DANNs can only model feature representations that are common across both domains. They also suffer from a vanishing gradient problem when the domain classifier correctly distinguishes source and target representations. (Shen et al., 2018) solve this problem in domain adaptation and was inspired from the work of (Arjovsky et al., 2017) on Wasserstein Generative Adversarial Networks. As shown in (Shen et al., 2018) domain adaptation using Wasserstein techniques is more stable than gradient reversal layers in terms of training. They try to minimize the estimated Wasserstein distance (also known as Earth Mover’s Distance) instead of developing a classifier to differentiate domains. According to a recent study on question pair classification, the two adversarial approaches (DANN and Wasserstein based) get equal results, but Wasserstein allows for more stable training (Shah et al., 2018).

Finding shared representations across the two domains has the disadvantage of leaving the shared representations sensitive to noise that is linked with the underlying common distribution. To solve this problem (Bousmalis et al., 2016) proposed domain separation networks (DSNs) to model features that pertain to both the source and target domains. In contrast to DANNs, DSNs add the concept of a private subspace for each domain that encapsulates domain-specific characteristics. The representations shared by the domains are captured in a common subspace, which is enforced by the use of autoencoders and explicit loss functions. Separate private encoders (one for each domain) and a common encoder are used by DSNs to separate latent representations. By locating a common subspace that is orthogonal to the private subspaces, the DSN approach is able to isolate the information

that is specific to each domain and create more meaningful representations for the task at hand. Note that this is a supervised technique and is comparable to the conventional supervised technique of (Daumé III, 2009).

The primary disadvantage of DSNs is that the decoder only uses domain-specific representations, leaving the classifier to be trained exclusively on domain-invariant representations. Further, their success has been limited to computer vision problems. To solve this (Shi et al., 2018) propose genre separation networks (GSNs) as a variation of DSNs incorporating a unique reconstruction component that uses both shared and private feature representations in the learning process.

Despite its popularity, a disadvantage of adversarial techniques, as pointed out in (Han and Eisenstein, 2019), is that they need careful balancing of objectives (Kim et al., 2017; for Computing Machinery, 1983) to prevent learning instability (Arjovsky et al., 2017).

**Re-weighting based Methods:** Re-weighting based methods are a type of instance-level domain adaptation. The intuition behind instance weighting (also known as significance weighting) is to give each training instance a weight proportional to how similar it is to the target domain (Jiang and Zhai, 2007). Instance weighting might be seen as an alternative to domain adversaries. While domain adversaries differentiate domains in a joint model to learn domain invariant representations, instance weighting decouples domain detection for a-priori weight estimation.

Maximum mean discrepancy (MMD)(Gretton et al., 2006) and its more efficient counterpart, kernel mean matching (KMM) (Gretton et al., 2009), are two methods that directly re-weight the loss depending on domain discrepancy information. In replicating a kernel Hilbert space, KMM re-weights the training instances so that the means of the training and test points are near to each other. Instance weighting was used for domain adaptation in NLP by (Jiang and Zhai, 2007), who recommended that weights be learned by first training domain classifiers. Despite early good results, the efficacy of re-weighting based methods in neural settings is yet to be determined. A preliminary research found no substantial improvements in part of

speech tagging (Plank et al., 2014).

#### 2.4.2 Data modification based methods

Due to the rapid growth of data and the popularity of pre-training methods, data-centric approaches have recently become more popular. Next, we'll go through the data-centric approaches, which differ depending on whether they utilize pseudo-labeling, choose relevant data, or use large amounts of unlabeled data or auxiliary tasks for model pre-training.

### **Pseudo Labeling**

The intuition behind pseudo-labeling is to use a trained classifier to predict labels on unlabeled examples, which are subsequently considered as ‘pseudo’ gold labels. Many such works use semi-supervised methods (Abney, 2007; Zhu and Goldberg, 2009) such as bootstrapping or temporal ensembling (Laine and Aila, 2018). Commonly used types of bootstrapping methods include self-training (Charniak, 1997; McClosky et al., 2006), co-training (Blum and Mitchell, 1998; Steedman et al., 2003), and tri-training (Zhou and Li, 2005; Søgaard and Rishøj, 2010; Saito et al., 2017). These methods either use the same model, a teacher model, or multiple bootstrap models, which may include slower but more accurate hand-crafted models (Petrov et al., 2010) to guide pseudo-labeling.

While most pseudo-labeling works date back to traditional non-neural learning methods, these have recently been revived for the neural world. For example, classic approaches like tri-training, provide a good foundation for domain shift in neural times (Ruder and Plank, 2018). Pseudo-labeling has been investigated for parsing with contextualized word representations (Rotman and Reichart, 2019; Lim et al., 2020) and adaptive ensembling (Desai et al., 2019) has been proposed as an extension of temporal ensembling.

### Data Selection

The goal of data selection is to find the best matching data for a new domain, which is usually done using perplexity (Moore and Lewis, 2010) or domain similarity metrics like Jensen-Shannon divergence over term or topic distributions (Plank and Van Noord, 2011). This is a relatively unexplored area for domain adaptation, but is gaining traction due to the onset of large pre-trained models especially to decide which data should they be trained on (Aharoni and Goldberg, 2020). This method has primarily been explored for machine translation (Axelrod et al., 2011), parsing (Ruder and Plank, 2017), and sentiment analysis, but only for supervised domain adaption settings. Distance metrics have recently been employed in a bandit-based strategy for multi-source domain adaptation of sentiment classification models (Guo et al., 2020). Another line of research investigates whether adapting big pre-trained models to the domain of a target task is still helpful, as well as the use of data selection to avoid expensive expert selection (Gururangan et al., 2020). In this work the authors propose two multi-phase pre-training techniques with promising results on text classification tasks.

### Pre-Training

In NLP, large pre-trained models have become popular (Howard and Ruder, 2018; Devlin et al., 2019; Peters et al., 2018). Fine-tuning a transformer-based model with a minimal quantity of labeled data has become a de-facto norm for NLP tasks. Starting with the pre-trained model weights, a new task-specific layer is trained using supervised data. However the question relevant to domain adaptation is whether domains and varieties are still relevant. In this section we will quickly review a few pre-training based techniques used for domain adaptation and address this question subsequently.

From a domain adaptation perspective, pre-training based methods can be classified into two: classic pre-training and adaptive pre-training. Classic Pre-training ((e.g., multilingual BERT; language-specific BERTs from scratch)) may be thought

of as a simple adaptation, similar to zero-shot in cross-linguistic learning. The main idea here is to train encoders with self-supervised and unsupervised objectives such as (masked) language models (Beltagy et al., 2019; Peters et al., 2018; Devlin et al., 2019).

Adaptive pre-training includes classic pre-training, followed by additional phases of pre-training on unlabeled data or labeled data from intermediate higher-resource auxiliary tasks. As shown in Han and Eisenstein (2019) adaptive pre-training is beneficial for domain shift. Here the authors show that in adaptive pre-training contextualized embeddings are modified to text from the target domain via masked language modeling.

Adaptive pre-training methods can be further divided into two types depending on whether unlabeled data or auxiliary labeled data (also known as intermediate tasks data) is employed. If data is available, these variations can be merged, and fine-tuning applies to all settings.

The first type Multi-phase pre-training, employs two or more phases of secondary pre-training, ranging from wide coverage to domain/task-adaptive pre-training (e.g., BioBERT, AdaptaBERT, DAPT, TAPT). They are distinguished by the type of unlabeled data they use. i.e., whether the data is broad-domain, domain-specific, or task-specific. For example, (Gururangan et al., 2020) proposes domain-adaptive pre-training (DAPT) from a broader corpus, and task-specific pre-training (TAPT), which uses unlabeled data closer to the task distribution. This implies that there is a range of domains with different levels of granularity, validating theories about domain similarity (Plank, 2011; Baldwin et al., 2013).

The second type of adaptive pre-training is auxiliary-task pre-training. It uses labeled auxiliary tasks via multi-task learning (MTL) (Peng and Dredze, 2016) or intermediate-task transfer (Phang et al., 2018, 2020). STILT (supplementary training on intermediate labeled-data tasks for transfer) was introduced by the latter (Phang et al., 2018), and this approach was recently used in cross-lingual learning, where English is utilized as an intermediate task for zero-shot transfer (Phang et al., 2020).

Another important topic to consider for pre-training is the choice of data used for pre-training (or auxiliary tasks). Currently, transformer models (Devlin et al., 2019) are trained on either huge general data sets, such as BookCorpus and Wikipedia in BERT, or target-specific samples, such as papers from Semantic Scholar in SciBERT (Beltagy et al., 2019) and PubMed abstracts and PMC full-text articles in BioBERT (Lee et al., 2020). However, it is still an open question as to what constitutes meaningful data. Today, it's either broad background information, domain-specific target data, or a combination of the two, with auxiliary tasks and intermediate training phases thrown in for good measure. The majority of them were hand-picked, pointing to interesting links between data selection and developing better curricula (Tsvetkov et al., 2016) to learn during domain shift (Ruder and Plank, 2017).

While huge pre-trained models have shown to be effective in domain adaptation, there are still numerous concerns and problems. Recent research has found that these models suffer when dealing with out-of-domain data, that maximum likelihood training renders them overconfident (Oren et al., 2019), and that calibration is crucial for out-of-domain generalization (Hendrycks et al., 2020). The brittleness of the process is another well-known problem with fine-tuning (Phang et al., 2018). As shown in (Dodge et al., 2020) different runs, even with the identical hyper-parameters, can provide radically different results, and training data order and seed selection have a significant influence.

#### 2.4.3 Methods that use both data centric and model centric approaches

Work on the confluence of data-centric and model-centric techniques include combining semi-supervised objectives with an adversarial loss (Lim et al., 2020; Alam et al., 2018), combining pivot-based approaches with pseudo-labeling (Cui and Bollegala, 2019) and more recently, contextualized word embeddings (Ben-David et al., 2020). Some other similar works include combining multi-task approaches with domain shift (Jia et al., 2019), multi-task learning with pseudo-labeling (multi-task tri-training) (Ruder and Plank, 2018). Further (Desai et al., 2019) uses a student-

teacher network with a consistency-based self-ensembling loss and a temporal curriculum to explore temporal and topic drift in political data categorization.

In conclusion, all these works show that in today’s models, domains are still important and that domain-relevant data is crucial for pre-training in both high and low resource setups. However, more research is needed into what these models capture and how they can be trained effectively in light of known test distributions or out-of-domain situations. Further, most of the domain adaptation methods mentioned above have been used for tasks like machine translation, sentiment analysis and parsing. To the best of our knowledge our work, detailed in the next few chapters, is the first attempt of using domain adaptation for RTE, especially for fact verification datasets.

## CHAPTER 3

## Towards the Necessity for Debiasing Natural Language Inference Datasets

As mentioned in the previous chapters, the task of natural language inference (NLI) is considered to be an integral part of natural language understanding (NLU). In this task, which can be seen as a particular instance of recognizing textual entailment (RTE) (Fyodorov et al., 2000; Condoravdi et al., 2003; Bos and Markert, 2005; MacCartney and Manning, 2009), a model is asked to classify if a given sentence (premise) *entails*, *contradicts* or is *neutral* given a second sentence (hypothesis).

In order to advance any task in natural language processing (NLP), quality data sets are quintessential. Some such data sets which have enabled the advancement of NLI (and fact verification) are SNLI (Bowman et al., 2015) MNLI (Williams et al., 2017), FEVER (Thorne et al., 2018a), and FNC (Pomerleau and Rao, 2017). However these datasets are not devoid of biases (subtle statistical patterns in a dataset, which could have been introduced either due to the methodology of data collection or due to an inherent social bias). For example, (Gururangan et al., 2018b) and (Poliak et al., 2018) show that biases were introduced into the MNLI dataset by certain language creation choices made by the crowd workers. Similarly, (Schuster et al., 2019) show that in the FEVER, the REFUTES label (same as the *contradicts* label mentioned above) highly correlates with the presence of negation phrases.

These biases can be readily exploited by neural networks (NNs), and thus have influence on performance. As an example, (Gururangan et al., 2018b) demonstrate that many state of the art methods in NLI could still achieve reasonable accuracies when trained with the hypothesis alone. Similarly, (Suntwal et al., 2019b) show that some NN methods in NLI with very high performance accuracies are heavily dependent on lexical information. Further (Yadav et al., 2019) show that this issue

is relevant not just in NLI but in other NLP applications such as question answering (QA) also.

This tendency of NNs to inadvertently exploit such dataset artifacts is likely worsened by the fact that currently the success of NLP approaches is almost exclusively measured by empirical performance on benchmark datasets. While this emphasis on performance has facilitated the development of practical solutions, they may lack guidance as they are often not motivated by more general linguistic principles or human intuition. This makes it difficult to accurately judge the degree to which these methods actually extract reasonable representations, correlate with human intuition or understand the underlying semantics (Dagan et al., 2013).

In this chapter we postulate that altering these datasets based on lexical importance is beneficial for organizing research and guiding future empirical work. While the technique of de-lexicalization (or masking) has been used before (Zeman and Resnik, 2008), we have expanded it by incorporating semantic information (the assumption that meaning arises from a set of independent and discrete semantic units) (Peyrard, 2019). Since these techniques are general and compatible with most existing semantic representations, we believe they can be further extended onto datasets used for other NLP tasks. Thus, by enabling integration of these techniques into the training pipeline, we hope to control lexicalization in the datasets which the NN methods possibly depend upon.

In particular, the contributions of this chapter are:

(1) To motivate further research in using de-lexicalized datasets, we present the de-lexicalized versions of several benchmark datasets used in NLI (e.g., FEVER, Fake News Challenge, SNLI, and MNLI), along with the corresponding software for the de-lexicalization. These datasets have been de-lexicalized using several strategies which are based on human intuition and underlying linguistic semantics. For example, in one of these techniques, lexical tokens are replaced or masked with indicators corresponding to their named entity class (Grishman and Sundheim, 1996). This differs from early works on de-lexicalization (Zeman and Resnik, 2008) in that we earmark overlapping entities (between the hypothesis and premise) with a unique

Config.	Claim	Evidence
Lexicalized	With Singapore Airlines, the Airbus A380 entered commercial service.	The A380 made its first flight on 27 April 2005 and entered commercial service on 25 October 2007 with Singapore Airlines.
OA-NER	With <code>organization-c1</code> , the <code>misc-c1</code> entered commercial service.	The A380 made its <code>ordinal-e1</code> flight on <code>date-e1</code> and entered commercial service on <code>date-e2</code> with <code>organization-c1</code> .
OA-NER+SS Tags	With <code>organization-c1</code> , the <code>artifact-c1</code> <code>motion-c1</code> commercial <code>act-c1</code> .	The A380 <code>stative</code> its <code>ordinal-e1</code> <code>cognition-e1</code> on <code>date-e1</code> and <code>motion-c1</code> commercial <code>act-c1</code> on <code>date-e4</code> with <code>organization-c1</code> .

Table 3.1: Example illustrating our masking techniques, compared to the original fully lexicalized data.

id. Further we also explore semantic lifting to WordNet synsets using Super Sense tags (Ciaramita and Johnson, 2003; Miller et al., 1990).

(2) We analyze and examine the effect of such de-lexicalization techniques on several state of the art methods in NLI and confirm that these methods can still achieve comparative performance in domain. We also show that in an out-of-domain set up, the model trained on de-lexicalized data outperforms that of the state of the art model trained on lexicalized data. This empirically supports our hypothesis that de-lexicalization is a necessary process for meaningful machine learning.

### 3.1 Masking Techniques

In (Suntwal et al., 2019b) the authors explore multiple methods for de-lexicalization. In this chapter we choose the two of their highest performing masking methods.

#### 3.1.1 Overlap-Aware Named Entity Recognition

In our overlap-aware named entity recognition (OA-NER) technique, the tokens in a given dataset are first tagged as named or numeric entities (NEs) by the named

Method	MNLI matched lexicalized	MNLI matched de- lexicalized	MNLI mis- matched lexicalized	MNLI mis- matched de- lexicalized
DA	60.9% ( $\pm 0.09$ )	64.5% ( $\pm 0.12$ )	61.4% ( $\pm 0.21$ )	64.8% ( $\pm 0.02$ )
ESIM	68.8% ( $\pm 0.01$ )	68.1% ( $\pm 0.11$ )	69.4% ( $\pm 0.21$ )	69.1% ( $\pm 0.23$ )

Table 3.2: Performance of various high performing NN methods over lexicalized and de-lexicalized versions of the same dataset. ‘Matched’ is the in-domain partition of the MNLI validation dataset, and ‘mis-matched’ is the out-of-domain partition. The performance of both the methods remain close to each other in de-lexicalized and lexicalized versions of the same dataset, which validates that our de-lexicalization techniques preserve the original information of the text.

entity recognizer (NER) of CoreNLP (Manning et al., 2014). Next, to capture the entity overlap between premise and hypothesis sentences, we uniquely enumerate the named entities. Specifically, in the claim (c) the first instance of an entity is tagged with **c1**. Subsequently wherever, in claim or evidence, this *same* entity is found next, it is replaced with this unique tag. In contrast, if an entity exists only in evidence, it is marked with an **e** tag. For example **person-c1** denotes the first time the proper name is found in claim, while **location-e3** indicates the third location found in evidence. An example of this is shown in Table 4.7.

### 3.1.2 OA-NER + Super Sense (SS) Tags

Our second type of masking relies on super sense tagging, a technique that uses sequence modeling to annotate phrases with their corresponding coarse WordNet senses (Ciaramita and Johnson, 2003; Miller et al., 1990). In this masking technique, in addition to the OA-NER replacement, other lexical items such as common nouns and verbs are replaced with their corresponding super sense tags. For example, as shown in Table 4.7, *Airbus A380* is replaced with *artifact* and *enter* is replaced with *motion* (more general abstractions captured by their WordNet hypernym synsets). Further unique overlap is also explicitly indicated, in the same manner as with OA-NER (see Table 4.7). This technique, specifically, will be used to explore the

impact of semantic lifting (i.e., if a more coarse-grained type is possibly less domain dependent) in both the in-domain and out-of-domain settings.

<b>Train Domain</b>	<b>MNLI</b>	<b>FEVER</b>	<b>FNC</b>
<b>Eval Domain</b>	<b>MedNLI</b>	<b>FNC</b>	<b>FEVER</b>
Lexicalized	51.4% ( $\pm 0.03$ )	48.8% ( $\pm 0.11$ )	41.1% ( $\pm 0.21$ )
OA-NER	51.5% ( $\pm 0.02$ )	53.5% ( $\pm 0.09$ )	46.4% ( $\pm 0.12$ )

Table 3.3: Performance accuracies of the Decomposable Attention against various masking techniques when tested out-of-domain. The accuracies reported are the mean across 3 random seed experiments using the Decomposable Attention model, with their corresponding standard deviations mentioned in brackets. The “Train Domain” row indicates the training datasets, while the “Eval Domain” indicates the domain of the corresponding evaluation partitions. For example, one experiment trained the DA method on FEVER and evaluated the resulting model on the testing partition of FNC (column 3).

### 3.2 Datasets and Methods

For analyzing the effects of various de-lexicalization operations we chose four popular datasets in NLI: Multi-genre NLI (MNLI) dataset (Williams et al., 2017), Fact Extraction and Verification (FEVER) dataset (Thorne et al., 2018a), the Fake News Challenge (FNC) dataset (Pomerleau and Rao, 2017), and the Medical NLI (MedNLI) dataset (Romanov and Shivade, 2018). In MNLI the trained models were tested on both of the validation partitions, the matched partition (which serves as the in-domain partition) and mismatched (the out-of-domain) partition.

We ran these experiments using two high-performing NLI methods: the Decomposable Attention (DA) (Parikh et al., 2016) and the Enhanced Sequential Inference (ESIM) (Chen et al., 2016).<sup>1</sup> All the methods were re-trained out of the box (without any parameter tuning) and tested on the corresponding evaluation partitions of the dataset.

<sup>1</sup>These models were the state-of-the art when these experiments were conducted. Further, these experiments are not meant to achieve or beat the state-of-the-art performance in-domain, but simply to prove the point that delexicalization helps. Further, the same is proven with another recent state of the art, Transformers in the next chapters

Masking strategy	FNC	FEVER
Lexicalized	68.9% ( $\pm 0.01$ )	83.4% ( $\pm 0.02$ )
OA-NER	65.8% ( $\pm 0.03$ )	82.3% ( $\pm 0.12$ )
OA-NER+SS	45.5% ( $\pm 0.29$ )	75.2% ( $\pm 0.21$ )

Table 3.4: Performance accuracies of the Decomposable Attention against various masking techniques when tested in-domain for FNC and FEVER datasets. The “Lexicalized” row shows the accuracies when DA was trained using the corresponding lexicalized data. This demonstrates that while de-lexicalization with OA-NER maintains the performance, the addition of Super Sense tags reduces the accuracy, emphasizing the fact that the amount of granularity to use is still an open problem. The accuracies reported are the mean across 3 random seed experiments using the Decomposable Attention model, with their corresponding standard deviations mentioned in brackets.

### 3.3 Results and Discussion

Table 3.2 shows the performance of each of these methods (DA and ESIM) on both the lexicalized and de-lexicalized versions of the MNLI dataset. As shown in the table, the accuracies of both the methods increase when trained with the de-lexicalized version of the dataset. This aligns with our intuition that de-lexicalization helps towards de-biasing these datasets, and thus preventing the NN methods from being *distracted* by statistical patterns that are not meaningful for the task at hand.

While the ability of a NN method to derive reasonable representations within the training domain is important, it is also important to have the ability to transfer across domains. Hence, to test the effect of de-lexicalization on domain transferability we picked one of the methods, decomposable attention (DA), trained it in one domain and tested it in an out-of-domain setting (the DA was chosen since it was provided off-the-shelf with FEVER baseline code). These results can be seen in Table 3.3. Specifically, the table shows the accuracies in three settings, i.e., when the model was trained on MNLI and then tested on MedNLI datasets, when it was trained on FNC and tested on the FEVER datasets, and when the model was trained on FEVER and tested on FNC datasets. Note that in some cases of out-of-domain

experiments, the label space of the source domain did not match with that of the target domain. Specifically, while the FEVER dataset consisted of data belonging to 3 classes, the FNC dataset had data points belonging to 4 classes. To enable us to evaluate using the official scoring measures of a target domain, we followed the label alignment approach used in (Suntwal et al., 2019b). For example, while the data points that belonged to the class *supports* were mapped to *agree*, and *refutes* to *disagree*, the ones in the class *not enough info* were further divided to align with the *unrelated* and *discuss* labels.

The experiments summarized in these tables highlight three observations: (a) The models trained on de-lexicalized data do not perform worse than the ones trained using lexicalized datasets; (b) in the two settings with texts where the named entities discussed are well covered by the NER used in this chapter (from FEVER to FNC, and from FNC to FEVER), the results demonstrate that the semantic lifting provided by our OA-NER method improves domain transfer considerably; and (c) we do not see a significant improvement in the transition from MNLI to MedNLI. We suspect this is because of the limited overlap of named entity types between the MedNLI and MNLI. For example while the named entity recognizer (CoreNLP) we used, focuses on PERSON, ORGANIZATION, etc. the MedNLI dataset contains more medical terms related diseases and symptoms. This possibly necessitates the importance of exploring using a domain-relevant NER.

Also this chapter touches upon questions about which granularity offers a good approximation of semantic meanings. For example, Table 3.4 shows the performance of Decomposable Attention against various de-lexicalization strategies. It can be seen that, while the addition of OA-NER tags does not change the performance significantly, the semantic lifting provided by the SS tags decreased the accuracies considerably in both cases (FEVER and FNC). This demonstrates that while generalizing away from lexical items is important, *how much* to generalize remains an open research problem.

Claim	Correct label	Lex model prediction	Delex model prediction	Bias in the lexicalized model	Most important entities
Did Argentina’s President adopt a jewish baby to stop it from becoming a werewolf ?	Discuss	Disagree	Discuss	Argentina - Disagree : 54.7% Argentina - Discuss : 45.2%	Lex model: [n’t, n’t, Argentina] Delex model: [only, MISCc1, LOCATIONc1]
Trump fired Comey over mishandling of Clinton emails.	Disagree	Agree	Disagree	Clinton - Agree : 63.5% Clinton - Disagree : 36.5%	Lex model: [fired, Clinton] Delex model: [Only, PERSONc1]
U.S. confirms authenticity of second journalist beheading video.	Discuss	Agree	Discuss	U.S - Discuss : 82.5% U.S - Agree : 17.5%	Lex model: [U.S.] Delex model: [LOCATIONc1]

Table 3.5: Data points in which the model trained on lexicalized data made an incorrect prediction while the model trained on the data de-lexicalized with OANER made the right prediction.

Class	Percentage Misclassified
Agree	46.2%
Discuss	31.8%
Disagree	11.6%
Unrelated	10.2%

Table 3.6: The percentage of labels that were mis-classified by the model that was trained on de-lexicalized data.

### 3.4 Qualitative Analysis

To better understand the merits of the proposed de-lexicalization techniques for bias neutralization, we sample several data points and bin them into two categories: data points that were misclassified by the lexicalized model but are classified correctly

	<b>Agree</b>	<b>Discuss</b>	<b>Disagree</b>	<b>Unrelated</b>
<b>Agree</b>	-	84.4%	14.1%	1.8%
<b>Discuss</b>	39.1%	-	29.3%	31.1%
<b>Disagree</b>	11.7%	79.4%	-	8.8%
<b>Unrelated</b>	10%	80%	10%	-

Table 3.7: Confusion matrix that highlights the distribution of the incorrectly predicted labels. Rows indicate the incorrectly predicted labels and columns indicate the corresponding original correct labels.

by the de-lexicalized model, and vice versa.

#### 3.4.1 Positive Changes

Table 3.5 shows several data points in which the model trained on lexicalized data made an incorrect prediction while the model trained on de-lexicalized data made the right prediction. As column three shows, the model trained on de-lexicalized data was able to overcome the bias, which possibly enabled the model trained on lexicalized data to make the incorrect prediction. For example, in the second data point the bias of *Clinton* towards the label *Agree* (i.e., the percentage of data points where the entity *Clinton* co-occurred with the label *Agree*) is 63.5%.

However, the model trained on de-lexicalized data was able to predict the label with a lower bias (*Disagree* with 36.5%). Further, column four shows the entities in which the corresponding model had placed the highest importance (as derived from their corresponding attention weights) for each of the trained models. In the first example it can be seen that while the model trained on lexicalized data considered the entity *Argentina* to be very important, the model trained on de-lexicalized data gives importance to the another overlap aware tag (MISCELLANEOUSc1) along with relegating the overlap aware tag of *Argentina* (LOCATIONc1) to lower importance. This indicates a possible de-coupling from the bias that the model has achieved, along with promoting other entities more important to the given task.

<b>Claim</b>	HP is better not together – company to split into enterprise and PC/printer businesses.
<b>Evidence</b>	HP’s home-focused and business divisions have frequently seemed at odds with each other, and apparently the company agrees. The Wall Street Journal claims that the tech giant is about to split into two companies, one focused on PCs and the other dedicated solely to corporate hardware and services. If the report is accurate, the separation could be announced as early as Monday. The exact reasoning behind the move hasn’t been mentioned, but the PC-centric group would be headed by one of its existing executives, Dion Weisler; current CEO Meg Whitman would run the business group and keep an eye on the other company by serving as its chairman of the board. However true the rumor may be, such a move wouldn’t be all that surprising – much of the computing industry has been restructuring and rescaling to cope with a world where the PC’s role is rapidly evolving . Source : Wall Street Journal
<b>Label</b>	Discuss

Table 3.8: An example of a data point whose original label was *discuss* and the model trained on overlap-aware de-lexicalized data predicted as *agree*.

	<b>Agree</b>	<b>Disagree</b>	<b>Discuss</b>	<b>Unrelated</b>
<b>PERSON-c1</b>	62.3%	21.5%	3.6%	12.4%
<b>DATE-e1</b>	57.7%	23.8%	3.1%	15.3%
<b>PERSON-e1</b>	52.5%	21.6%	4.8%	22.2%
<b>Original label distribution</b>	55.5%	20.4%	4.1%	20.3%

Table 3.9: Relative percentages of some OA-NER tags with respect to their labels along with the original label distribution in the training data.

### 3.4.2 Negative Changes

Similar to the above task, but to better understand the limits of the proposed de-lexicalization techniques, we sampled data points where the model trained on

de-lexicalized data made incorrect predictions. The distribution of such incorrectly predicted labels, is given in table 3.6. As seen from this table, of all the incorrect predictions made, the maximum were of the label *agree* (46.3%) followed by the labels *discuss*, *disagree* and *unrelated*.

In table 3.7 we show the distribution of the incorrectly predicted labels against their original labels, in the form of a confusion matrix. The table focuses solely on the data points that are misclassified, so, unlike standard confusion matrices, the diagonal is empty here.

A case of particular interest in this analysis are the high percentages of wrongly predicted labels (*agree* (84.4%), *disagree* (79.1%), *unrelated* (80%)) when the original correct label was *discuss*. This suggests that the complexity and subtlety of language understanding gets particularly pronounced in cases of classifying neutral (*discuss*) texts. An example of such an incorrectly classified data point is presented in Table 3.8.

In table 3.9 we show the relative percentages of the OA-NER tags with respect to their classes along with the original label distribution in the training data. While the previous analyses indicate that we did overcome some biases, this table suggests that we have created new ones, which in-turn affect the quality of the model. This highlights that precisely determining the *right* amount of granularity needed for de-lexicalization to minimize bias remains an open research problem.

### 3.5 Conclusion

In this chapter we investigated the need for de-lexicalization techniques to reduce the potential dependence of NN methods on lexicalized items in NLI tasks. We specifically explore two masking techniques to de-lexicalize datasets: the first one replaces the lexical entities in an overlap-aware manner, and the second one additionally incorporates semantic lifting of nouns and verbs.

Our experiments show that de-lexicalization achieves comparative results *in-domain* with the state of the art methods trained on lexicalized data. Importantly, we show that methods trained on de-lexicalized data transfer considerably better

*out-of-domain*, which confirms the importance of de-lexicalization in NLI tasks for domain transferability. While there is still room for exploration in de-lexicalization techniques, we present this methodology as a means to conduct more meaningful machine learning experiments. To facilitate this, we release the de-lexicalized versions of MNLI, FEVER and FNC datasets.

One future direction of interest is exploring the right level of masking granularity needed for de-lexicalization, which is likely dependent on the task at hand. In addition, we would also like to explore combining de-lexicalization with knowledge distillation to transfer learning across domains.

### 3.6 Language Resource References

All the software and masked datasets for our proposed approach are open-source and publicly available. The datasets are available at

[https://osf.io/szdkn/?view\\_only=4845641a80624ac493ca14df34e68e8c](https://osf.io/szdkn/?view_only=4845641a80624ac493ca14df34e68e8c)

and the code is available on GitHub at:

<https://github.com/clulab/releases/tree/master/lrec2020-masking>.

## CHAPTER 4

### Using Data Distillation To Achieve Domain Transfer

Neural networks (NNs) play a key role in most modern natural language processing (NLP) systems, obtaining state-of-the-art performance (Devlin et al., 2018; Sun et al., 2018; Bohnet et al., 2018) in many complex tasks, for example in recognizing textual entailment (Kim et al., 2018), fake news detection (Baird et al., 2017) and fact verification (Nie et al., 2018). However, these models depend heavily on lexical information that may transfer poorly between different domains. For example, in early experiments in the fact verification space, we observed that out of all the 34 statements containing the phrase “American Author”, 31 (91%) belonged to one class label. Such information could be meaningful say, in the literature domain, but transfers poorly to other domains such as science or entertainment. In this chapter our focus is to: (a) understand and estimate the importance that a neural network assigns to various aspects of the data while learning and making predictions, and (b) learn how to control for unnecessary lexicalization.

#### 4.1 Need for domain transfer in fact verification

As described before in chapter 2, Recognizing Textual Entailment (RTE) is the task of determining if one piece of text can be plausibly inferred from another. Fact verification is a sub task of RTE where the claim-evidence pairs are derived from more real life datasets like Wikipedia and News Articles. Neural networks have achieved state-of-the-art performance in several fact verification tasks and datasets. However, these learning methods have an assumption that the training and test sets have the same underlying distribution. But this is not true in real life scenarios. In several cases, the test datasets do not have annotated labels, and hence rely on

	<b>Accuracy when doing an in-domain evaluation in FEVER dataset</b>	<b>Accuracy when doing a cross-domain evaluation in FNC dataset</b>
State of the art model (BERT)	94.1%(±0.01)	68.9%(±0.07)

Table 4.1: Performance of a state of the art model, BERT, that was trained in one domain, drops considerably when tested on a related domain. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets.

models that are trained on other similar domains for predictions in this domain. As a result, performance on the target suffers, undermining models’ capacity to properly generalize in the wild. For example, as shown in Table 4.1, when a neural network model which produced state of the art performance (94.1%) in one fact verification dataset called FEVER (Thorne et al., 2018b), was used to predict and test on another related dataset, the Fake News Challenge (FNC) dataset (Pomerleau and Rao, 2017), the performance drops to 68.9%. Hence it is important to devise methods and techniques that can train models which perform well across similar domains.

## 4.2 Overview

The rest of this chapter is organized as follows. In section 4.3 we will describe the setup used in all the experiments in this chapter. Then, in section 4.4 we describe in detail the model and datasets used in our experiments. In section 4.5 we describe how we investigate the attention weights to confirm that the proposed problem does indeed exist in the setup we use. Further, we propose a few solutions to solve this problem in section 4.6. Section 4.7 shows the results of our experiments. We do an error analysis of these results in section 4.8 and 4.9 concludes this chapter.

### 4.3 Setup

In our experiments we first show that the aforementioned drop in performance is true in a cross-domain setting, and further suggest solutions to improve it. For this we use two distinct fact verification datasets for our experiments. The first one called the FEVER dataset was the dataset that was used in the 2018 Fact Extraction and Verification challenge (Thorne et al., 2018a). The second dataset we use is FNC, which was the dataset used for the Fake News Challenge (Pomerleau and Rao, 2017). The reason we chose these two datasets is that they are at the right balance of being in similar and dissimilar domains. They are both similar because the claim-evidence pairs in both datasets are taken from real-life datasets. For example, while the claim-evidence pairs in FEVER are chosen from Wikipedia, the same in FNC are chosen from news articles of the past few years. Hence they both deal with similar data: knowledge about real life incidents and people. However, they are dissimilar because Wikipedia and News Articles have different styles of prose. Further, the evidence in FNC is comprised of longer text since it is made from a combination of multiple sentences from news articles. A few examples of each can be found in Table 4.2 and Table 4.3.

In our experiments we first train on the training partition of one dataset, and use the same trained model to predict and evaluate on the test partition of the other dataset. Then we run the same experiments in the other direction. For example, in our experiments we first train on the training partition of FEVER, and use the same trained model to predict and evaluate on the test partition of FNC and vice versa. This experimental setup is shown in Figure 4.1. Note that this is the same setup that will be used in the other experiments in the rest of the chapters.

In FEVER the claim-evidence pairs are classified into 3 classes: *agree*, *disagree* and *not enough info*. In FNC, the claim-evidence pairs are classified into 4 classes: *agree*, *disagree*, *discuss* and *unrelated*. In order to evaluate models in a cross-domain setting, we modified the label space of the source domain to match that of the target domain. This also allows us to evaluate using the official scoring measures of the

Claim	Evidence	Label
With Singapore Airlines, the Airbus A380 entered commercial service.	The A380 made its first flight on 27 April 2005 and entered commercial service on 25 October 2007 with Singapore Airlines.	Agree
Colin Kaepernick became a starting quarterback during the 49ers 63rd season in the National Football League .	The community has the name of John Howard , a local pioneer .	unrelated
Stranger Things is set in Bloomington , Indiana.	Set in the fictional town of Hawkins, Indiana in the 1980s , the first season focuses on the investigation into the disappearance of a young boy by his friends, older brother and traumatized mother and the local police chief, amid supernatural events occurring around the town including the appearance of a psychokinetic girl who helps the missing boy 's friends in their own search.	Disagree
John Wick : Chapter 2 was theatrically released in the Oregon.	John Wick -LRB- whistleblower -RRB- , former Special Air Service major who was a source revealing a 2009 political scandal , see United Kingdom parliamentary expenses scandal.	Discuss

Table 4.2: Examples of claim and evidence sentences from FEVER dataset (Thorne et al., 2018a)

target domain. In particular, when training on FEVER and testing on FNC, the data points in FEVER that belong to the class *supports* were relabeled as *agree*, and those in *refutes* as *disagree*. The data points belonging to the third class *not enough info* were divided into *discuss* and *unrelated*. Details of this label matching is further elaborated in detail in section 4.4.1.

Next, we try to understand what causes neural networks to have a low performance in such a cross-domain setting. We hypothesize that the root cause of neural networks having low performance in a cross-domain setting is because they generally learn from lexical information in a given dataset, which transfer poorly

Claim	Evidence	Label
Pope Francis turns out not to have made pets in heaven comment.	NEW YORK — Pope Francis has given hope to gays , unmarried couples and advocates of the Big Bang theory. Now, he has endeared himself to dog lovers, animal-rights activists and vegans . Trying to console a little boy whose dog had died, Pope Francis told him in a recent public appearance on St Peter’s Square that “paradise is open to all of God’s creatures”. While...	Disagree
Back-from-the-dead Catholic priest claims God is a female	A 71 years old cleric Father John Micheal O’neal who was officially dead for more than 48 minutes, was re-started by medics. He claims he went to heaven and met God, which he describes as a warm and comforting motherly figure. Father...	Agree
Soon Marijuana May Lead to Ticket, Not Arrest, in New York.	After campaigning on a promise to reform stop-and-frisk, Mayor Bill de Blasio is set to launch his most significant effort to address the issues raised by the policy. Law-enforcement...	Not Enough Info

Table 4.3: Examples of claim and evidence sentences from FNC dataset (Pomerleau and Rao, 2017). Refer appendix for full text of evidence sentences.

between domains. To confirm this, we investigate the importance that a model assigns to various aspects of data while learning and making predictions, specifically, in a recognizing textual entailment (RTE) task. By inspecting the attention weights assigned by the model, we confirm that the part of speech tags which get the highest percentage of attention weights are the noun phrases. This confirms our hypothesis that the model relies on semantic patterns to make predictions.

Next, to mitigate this dependence on lexicalized information, we experiment with several strategies of masking. In one embodiment of such a technique, we replace named entities with their corresponding semantic tags along with a unique identifier to indicate lexical overlap between claim and evidence. In another, we replace other word classes in the sentence (nouns, verbs, adjectives, and adverbs) with their super sense tags (Ciaramita and Johnson, 2003). Our results show that, while performance on the in-domain dataset remains at par with that of the model trained on fully lexicalized data, it improves considerably when tested out of domain.

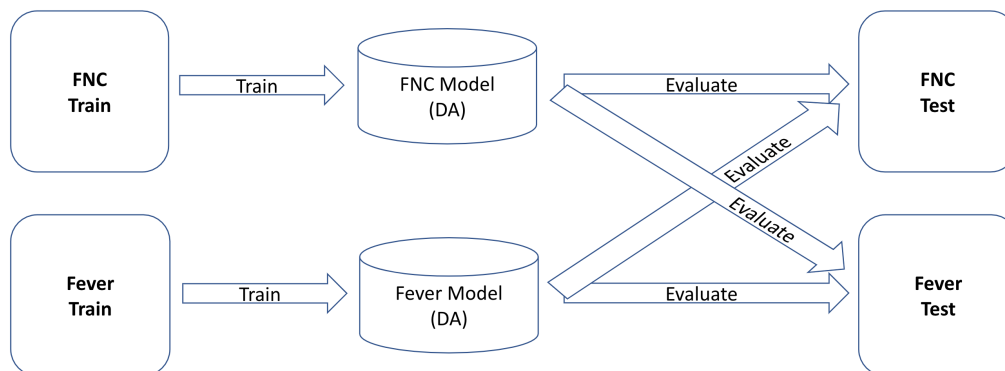


Figure 4.1: Experiment setup for testing the hypothesis that state of the art models drop in performance when tested on a dataset it was not trained on. In our experiments we first train on the training partition of one dataset (e.g.,FEVER), and use the same trained model to predict and evaluate on the test partition of the same dataset (e.g.,FEVER). The accuracy achieved here is considered as in-domain performance. We also use the same model to test on the test partition of the other dataset (e.g.,FNC). The accuracy achieved here is considered as the cross-domain performance of the corresponding model. Then all these experiments are run in the other direction i.e with the datasets swapped from first experiment.

For example, the performance of a state-of-the-art model trained on the masked Fake News Challenge (Pomerleau and Rao, 2017) data and evaluated on Fact Extraction and Verification (Thorne et al., 2018a) data improved by over 10% in accuracy score compared to the fully lexicalized model.

A known shortcoming of this setup is that it is only a limited approximation of real world scenarios. However, to the best of our knowledge, this is the first time such an experimental setup has been done for fact verification.

#### 4.4 Datasets and models

In this section we will describe in detail the data and models used for various experiments.

Dataset	Support	Refute	Discuss	Unrelated
FNC	5,581	1,537	13,373	54,894
FEVER	86,701	36,441	42,305	( <i>NEI</i> )

Table 4.4: Label distribution for the FEVER and FNC datasets. We consider the *agree* and *disagree* FNC labels as equivalent to the *support* and *refute* labels in FEVER. The FNC *not enough info (NEI)* label is listed below the more fine-grained *discuss* and *unrelated* FEVER labels.

#### 4.4.1 Datasets

As mentioned in the section 4.3 we use two distinct fact verification datasets for our experiments, FEVER and FNC.

##### **Fact Extraction and Verification (FEVER)**

The FEVER (Thorne et al., 2018a) dataset consists of 145,449 training data points, each of which has a claim and a set of evidences retrieved from Wikipedia using a baseline information retrieval (IR) module. The FEVER claim-evidence pairs were assigned labels from three classes: *supports*, *refutes*, and *not enough info (NEI)*.

Even though the partition of the FEVER dataset that was used in the final shared task competition was released publicly, the gold test labels were not. Hence we used the development portion (19,998 data points) instead as our test partition. We created our own development partition by randomly dividing the training partition into 80% for training (119,197 data points) and 20% (26,252 data points) for development. The evidence for data points that had the gold label of *not enough info* can be retrieved (using a task-provided IR component) either by finding the nearest neighbor to the claim or randomly (Thorne et al., 2018a).

##### **Fake News Challenge (FNC)**

The FNC dataset (Pomerleau and Rao, 2017) contains four classes (*agree*, *disagree*, *discuss*, and *unrelated*) and has publicly available training (49,972 data points) and test partitions (25,413 data points). Each data point consists of a claim and a set of evidence sentences. We split the training dataset into two partitions, with 40,904

records for training and 9,068 records for development.

### Cross-domain Labels

In order to evaluate models in a cross-domain setting, we modified the label space of the source domain to match that of the target domain. This also allows us to evaluate using the official scoring measures of the target domain. In particular, when training on FEVER and testing on FNC, the data points in FEVER that belong to the class *supports* were relabeled as *agree*, and those in *refutes* as *disagree*. The data points belonging to the third class *NEI* were divided into *discuss* and *unrelated* as follows. In the FEVER dataset, two separate methods were used to retrieve the evidence sentences for the class *NEI*: 1) sampling a sentence from the nearest page to the claim as evidence using their document retrieval component, and 2) sampling a sentence from Wikipedia uniformly at random. The evidence sentences retrieved using the nearest page technique were assigned the label *discuss* (since it was more likely to be topically relevant to the claim), and the rest were assigned the label *unrelated*. Also the distribution of labels was made similar to that of FNC. In particular, 16.8% of the total 35,639 data points that originally belonged to the class *NEI* in FEVER were thus assigned the label *discuss* and the rest (83.1%) were assigned the label *unrelated*. For the experiments in the opposite direction, i.e., when training on FNC and testing on FEVER, we collapsed the *unrelated* and *discuss* classes into a single class, *not enough info*. Please refer to Table 4.4 for a summary of label distributions.

#### 4.4.2 Models

For all of the experiments in this chapter we use the Decomposable Attention (DA) model (Parikh et al., 2016), which at the time of these experiments consistently achieved near state-of-the-art performance on RTE tasks<sup>1</sup>. In particular, we use

---

<sup>1</sup>This experiment was initially setup during the 2018 FEVER shared task which had provided Decomposable Attention as a strong baseline. At the completion of the shared task, most of the winning teams used another model (Chen et al., 2016). However, the aim of this work is not to outperform the state of the art performance, but rather to generalize to new domains without

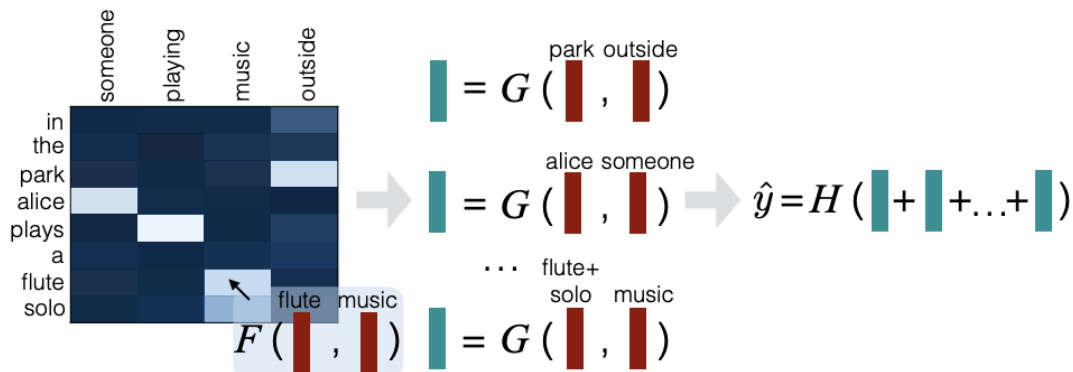


Figure 4.2: Pictorial overview of the Decomposable Attention approach, showing the Attend (left), Compare (center) and Aggregate (right) steps. Specifically,  $F$  denotes the a function in the attend step using which the unnormalized attention weights are obtained.

the AllenNLP<sup>2</sup> implementation of DA, which was provided by the FEVER task organizers. To investigate DA’s reliance on lexical information, we first examine its word-level attention weights (Section 4.5). Informed by these results, we propose several methods to mask the lexicalized data (Section 4.6), and evaluate them in in-domain and out-of-domain settings (Section 4.7). Since it is the quintessential part of all the experiments in this chapter, we will take a moment to briefly describe the functionalities of the decomposable attention model.

### Decomposable Attention

The Decomposable attention model was first proposed in (Parikh et al., 2016). This method solves the natural language inference problem by decomposing it into sub-problems that can then be handled individually using the attention mechanism (Bahdanau et al., 2014) which learns how to softly align two texts. The intuition behind this approach is that for natural language inference, merely aligning pieces of local text substructure and then aggregating this information is frequently sufficient. For example consider the following example sentences used for natural language retraining.

<sup>2</sup><https://github.com/allenai/allennlp>

inference (NLI).

1. Bob is in his room, but because of the thunder and lightning outside, he cannot sleep.
2. Bob is awake.
3. It is sunny outside.

The authors argue that inference between first two sentences can be achieved using alignment. First they align *Bob* in the first sentence with *Bob* in the second sentence. Then they align *cannot sleep* in the first sentence with *awake* in the second sentence. By understanding that these two are synonyms, it is now very straightforward to deduce that the second statement can be inferred/entails from the first. Similarly, aligning *thunder and lightning* with *sunny* and understanding that they are most likely *contradictory* will lead to the conclusion that *It is sunny outside* contradicts with the first sentence. This intuition is then used to create a more straightforward and lightweight method for NLI inside a neural framework called the decomposable attention (DA) method. Unlike other methods, DA depends just on alignment and is entirely decomposable in terms of the input text. Figure 4.2 depicts a summary of this methodology.<sup>3</sup>

Specifically, there are 4 components/steps to the proposed architecture: **embed**, **attend**, **compare** and **aggregate**. Given two phrases, in the **embed** step, all the words are mapped to their corresponding word vector representations. Next, in the **attend** step a soft alignment matrix is built using neural attention (Bahdanau et al., 2014). In this matrix each word is represented by an embedding vector as mentioned before. In the **compare** step the task is then decomposed into sub-problems that are solved individually using the (soft) alignment. Finally, in the **aggregate** step the findings of these sub-problems are then combined to get the final classification. Prior to the alignment stage, intra-sentence attention (Cheng et al., 2016) is also used to provide the model with a deeper encoding of substructures.

---

<sup>3</sup>this is an exact replica of the figure from the original work

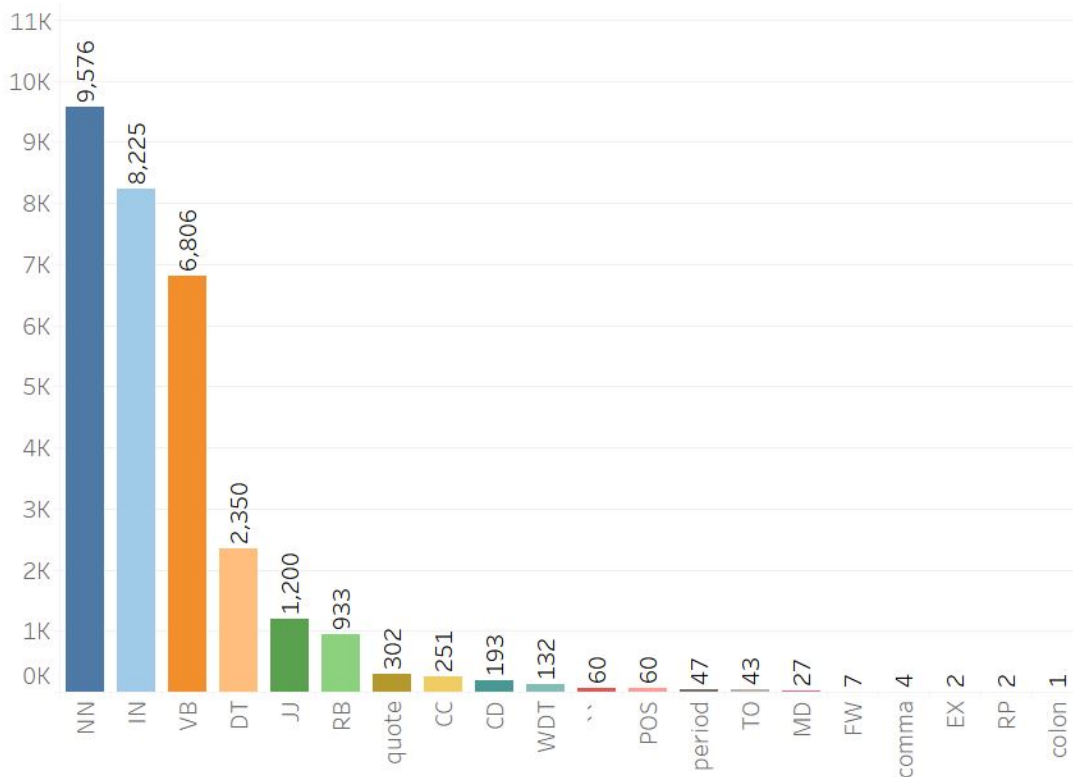


Figure 4.3: Distribution of POS tags that were assigned the highest attention weights by DA for incorrectly classified cross-domain examples.

Note that each embedding vector is normalized to a l2 norm of 1 and projected down to 200 dimensions, which is determined by hyper-parameter tweaking. Out-of-vocabulary words are hashed into one of 100 random embeddings, each with a mean of 0 and a standard deviation of 1. During training, all embeddings stay fixed, but the projection matrix is trained. The weights of all other parameters (hidden layers, etc.) are calculated using random Gaussians with a mean of 0 and a standard deviation of 0.01.

#### 4.5 Attention analysis

To understand and verify that models trained on lexicalized data transfer poorly in our tests we look at the attention weights the model assigns when it makes erroneous predictions. Specifically, we use the word-level attention weights (Bahdanau et al., 2014) learned by DA to perform an error analysis in the cross-domain evaluation. In

particular, we first trained two equivalent DA models, one on FEVER and the other on FNC. Next, we used both models to predict instances in FEVER development. For each data instance, we calculated the cumulative attention weights assigned by each of the two models to all evidence words. For each of the data instances that were *incorrectly* classified by the model trained out of domain (i.e., on FNC) and were *correctly* classified by the in-domain model, we selected the words that were present in the set of top three words with the highest attention score according to the out-of-domain model, but were *not* present in the equivalent set produced by the in-domain model. Such words indicate potential over-fitting of the out-of-domain model. Figure 4.3 shows the distribution of part-of-speech (POS) tags for these words. The figure indicates that, for incorrectly classified examples in cross-domain, the higher attention weights were assigned to nouns. Importantly, 43.1% of these noun phrases in FEVER are named entities. With this analysis, we are able to confirm that most of the weight is assigned to POS tags of nouns or elements of noun phrases, which confirms our observation that these models anchor themselves on lexical information that is more likely to be domain dependent.

#### 4.6 Delexicalization

As shown in 4.5 we found that the DA model assigns high attention weights to noun phrases. To mitigate the potential domain dependence introduced by these large attention weights, we propose several semantic masking techniques, which we compare with a deletion baseline. In these delexicalization methods, lexical tokens are substituted (or masked) with class markers (e.g., the corresponding named entity types). Examples of each form of masking are shown in Table 4.7.

Note that while delexicalization/masking has been utilized previously (Zeman and Resnik, 2008), we extend it here by integrating semantic information. Further note that the techniques we propose here are agnostic to the taxonomy used for delexicalization. However, for the pragmatic purposes we rely on existing tools, for example the named entity recognition tools available in open domain. Next we will describe the tools that were used for named entity recognition followed by the details

<b>Input Sentences</b>	<b>Named Entities</b>	<b>Named Entity Types</b>
Chris Manning teaches at Stanford University. He lives in the Bay Area.	Chris Manning, Stanford University, the Bay Area	Chris Manning: PERSON Stanford University: ORG the Bay Area: LOC

Table 4.5: An example of the named entities as found by the Stanford CORENLP named entity recognizer.

of the masking techniques devised using them.

#### 4.6.1 Named Entity Recognition Tools

Entity Recognition (ER) is a type of data extraction that aims to find sections of text (mentions) that match to entities and classify them into a pre-defined set of classes. Entity recognizers are useful in a variety of situations. Many connection extraction pipelines, for example, begin by identifying a relation’s probable arguments in a phrase using entity recognition, and then categorize or extract the relation type. Naturally, the sorts of arguments have a role in deciding whether relationship holds (if any). In this section we will describe the named entity recognition tools from open domain that we use for the data pre-processing step of our delexicalization/masking techniques.

#### **CoreNLP**

Stanford CoreNLP toolkit (Manning et al., 2014) is an expandable pipeline used for natural language analysis. It provides tools for the majority of the typical core natural language processing (NLP) steps from tokenization to co-reference resolution. Specifically, we use the named entity recognition sub module of the CoreNLP toolkit to recognize the named entities in each claim-evidence pairs. The named entity annotator recognizes 12 classes consisting of named (PERSON, LOCATION, ORGANIZATION, MISC), numerical (MONEY, NUMBER, ORDINAL, PERCENT), and temporal (DATE, TIME, DURATION, SET) entities. This annotator labels items primarily using one or more machine learning sequence models, but it may

<b>Input Sentence</b>	<b>Named Entities</b>	<b>Concept Types</b>
I don't think he's afraid to take a strong stand on gun control, what with his upbringing in El Paso.	think, he's, take_a_strong_stand, gun_control, upbringing, El_Paso	think: cognition he's :stative take_a_strong_stand :cognition gun_control :ARTIFACT, upbringing :ATTRIBUTE, El_Paso :LOCATION

Table 4.6: An example of identifying and labeling several lexical expressions with supersenses (Schneider and Smith, 2015) in a given sentence. Note that uppercase is used to denote nouns, while lowercase is used to denote verbs.

additionally invoke specialised rule-based components, such as those for labeling and interpreting times and dates. For example, Named entities are recognized using a combination of CRF sequence taggers Finkel and Manning (2009) trained on various corpora including CoNLL, ACE, MUC, and ERE corpora. Numerical entities are recognized using two rule-based systems, one for money and numbers and a separate state-of-the-art system for processing temporal expressions (Chang and Manning). An example of the output produced by this named entity recognition tool can be found in Table 4.5.

### Super Sense Tagger

Super sense tags were introduced in (Schneider and Smith, 2015) for detecting and semantically categorizing lexical phrases in running text. This was developed while solving the primary issue in computational lexical semantics for text corpora, which is to build and use abstractions that describe word meanings beyond what can be deduced from the orthography. As a solution the authors proposed that coarse-grained semantic classes provide a more portable, scalable, and linguistically grounded way to express lexical meanings. Thus they created 41 super senses for this purpose. These super sense are similar to the kinds used for named things (PERSON, LOCATION, etc.), but they are more broad, having semantic categories that apply to common nouns and verbs. Examples of such super sense tags devised for nouns can

Noun		
GROUP	1469	<i>place</i>
PERSON	1202	<i>people</i>
ARTIFACT	971	<i>car</i>
COGNITION	771	<i>way</i>
FOOD	766	<i>food</i>
ACT	700	<i>service</i>
LOCATION	638	<i>area</i>
TIME	530	<i>day</i>
EVENT	431	<i>experience</i>
COMMUNIC.*	417	<i>review</i>
POSSESSION	339	<i>price</i>
ATTRIBUTE	205	<i>quality</i>
QUANTITY	102	<i>amount</i>
ANIMAL	88	<i>dog</i>
BODY	87	<i>hair</i>
STATE	56	<i>pain</i>
NATURAL OBJ.	54	<i>flower</i>
RELATION	35	<i>portion</i>
SUBSTANCE	34	<i>oil</i>
FEELING	34	<i>discomfort</i>
PROCESS	28	<i>process</i>
MOTIVE	25	<i>reason</i>
PHENOMENON	23	<i>result</i>
SHAPE	6	<i>square</i>
PLANT	5	<i>tree</i>
OTHER	2	<i>stuff</i>

Verb		
STATIVE	2922	<i>is</i>
COGNITION	1093	<i>know</i>
COMMUNIC.*	974	<i>recommend</i>
SOCIAL	944	<i>use</i>
MOTION	602	<i>go</i>
POSSESSION	309	<i>pay</i>
CHANGE	274	<i>fix</i>
EMOTION	249	<i>love</i>
PERCEPTION	143	<i>see</i>
CONSUMPTION	93	<i>have</i>
BODY	82	<i>get...done</i>
CREATION	64	<i>cook</i>
CONTACT	46	<i>put</i>
COMPETITION	11	<i>win</i>
WEATHER	0	—

Figure 4.4: Examples of super sense categories for nouns and verbs

be found in figure 4.4.<sup>4</sup> To incorporate this super sense tagging into our masking techniques, we utilize the super sense tagging module<sup>5</sup> made publicly available by the authors. An example of the output produced by this module is also shown in table 4.6.

#### 4.6.2 Masking Techniques

As mentioned above, to mitigate the over dependency of neural network on semantic patterns, we propose a few masking techniques. The open source named entity recognizers mentioned above are used to create various masking techniques, which are described in detail below. An example of each of the masking/delexicalization<sup>6</sup> techniques mentioned below can be found in table 4.7.

Note that in all the techniques below, the first step is when both the claim and evidence sentences are passed through the CoreNLP’s named entity recognizer (Manning et al., 2014). Each of these techniques differ from each other based on

<sup>4</sup>Excerpt from the original work.

<sup>5</sup><https://github.com/nschneid/pysupersensetagger>

<sup>6</sup>Since most of these techniques involves replacing a lexicalized named entity with its type, we also call these masking techniques as de-lexicalization.

<b>Configuration</b>	<b>Claim</b>	<b>Evidence</b>
Lexicalized	With Singapore Airlines, the Airbus A380 entered commercial service.	The A380 made its first flight on 27 April 2005 and entered commercial service on 25 October 2007 with Singapore Airlines.
NE Deletion	With , the entered commercial service.	The A380 made its flight on and entered commercial service on with.
Basic NER	With <code>organization</code> , the <code>miscellaneous</code> entered commercial service.	The A380 made its <code>ordinal</code> flight on <code>date</code> and entered commercial service on <code>date</code> with <code>organization</code> .
OA-NER	With <code>organization-c1</code> , the <code>misc-c1</code> entered commercial service.	The A380 made its <code>ordinal-e1</code> flight on <code>date-e1</code> and entered commercial service on <code>date-e2</code> with <code>organization-c1</code> .
OA-NER + SS Tags	With <code>organization-c1</code> , the <code>artifact-c1</code> <code>motion-c1</code> commercial <code>act-c1</code> .	The A380 <code>stative</code> its <code>ordinal-e1</code> <code>cognition-e1</code> on <code>date-e1</code> <code>date-e2</code> <code>date-e3</code> and <code>motion-c1</code> commercial <code>act-c1</code> on <code>date-e4</code> <code>date-e5</code> <code>date-e6</code> with <code>organization-c1</code> .

Table 4.7: Example illustrating our various masking techniques, compared to the original fully lexicalized data (first row). Note that the masking tags were generated with real-world (imperfect) tools. For example, “Airbus A380” in the claim was correctly classified as `miscellaneous` by the NER tool, while “A380” in the evidence was not, thus preventing us from taking advantage of the overlap.

the subsequent steps.

### Named Entity (NE) Deletion Baseline

After the step of named entity tagging by CoreNLP, in this technique, each of the named entities (NEs) are deleted from the sentences. Since this is a data destruction technique, this will be considered as a baseline to compare the rest of the techniques against.

### Basic NER

In this technique the token sequences which are labeled as NERs are replaced by their corresponding label, e.g., `location`, `person`.

### Overlap Aware NER (OA-NER)

This technique additionally captures the *lexical overlap* between the claim and evidence sentences with entity ids. That is, the first instance of a given entity in the claim is tagged with `c1`, where the `c` denotes the fact that it was found in the claim sentence (e.g., `person-c1`). Wherever this *same* entity is found later, in claim or in evidence, it is replaced with this unique tag. If an entity is found only in evidence, then it is denoted by an `e` tag. (e.g., `location-e3` would be the third location found only in the evidence).

For example, in the claim-evidence pair shown in Table 4.7, when the named entity *Singapore Airlines* appears in the claim it is replaced with `organization-c1`, since it is the first `organization` to appear in claim. The same id is used wherever the same entity is seen again, e.g., in the evidence sentence. However, the date *27 April 2005* occurs only in the evidence, and hence it is replaced with `date-e1`. Importantly, we create pseudo-pretrained embeddings for these new OA-NER-based tokens by adding a small amount of random Gaussian noise (mean 0 and variance of 0.1) to pre-trained embeddings (Pennington et al., 2014) of the root word corresponding to the category (e.g., *person*). Thus the embeddings of all the sub-tags, while being unique, are close to that of the root word.

### Overlap Aware NER (OA-NER) + Super Sense (SS) Tags

As discussed before, super-sense tagging is a sequence modeling approach that annotates phrases with coarse WordNet senses (Ciaramita and Johnson, 2003; Miller et al., 1990). In this masking method, we not only replace named entities with their OA-NER tags mentioned above, but also replace other lexical items with their corresponding super sense tags, if found. As with the OA-NER approach, the lexical

	Configuration				
	Train Domain	FNC	FEVER	FEVER	FNC
Eval Domain	FNC	FNC	FEVER	FEVER	FEVER
Masking					
Lexicalized	96.3%( $\pm 0.18$ )	48.8% ( $\pm 0.01$ )	83.4%( $\pm 0.03$ )	41.1%( $\pm 0.29$ )	
Deletion	66.5%( $\pm 0.11$ )	40.3% ( $\pm 0.02$ )	75.4%( $\pm 0.07$ )	33.3% ( $\pm 0.14$ )	
Basic NER	69.4%( $\pm 0.04$ )	46.7% ( $\pm 0.11$ )	76.2%( $\pm 0.02$ )	35.7%( $\pm 0.13$ )	
<b>OA-NER</b>	65.5%( $\pm 0.11$ )	<b>53.5%</b> ( $\pm 0.09$ )	82.1%( $\pm 0.07$ )	46.4%( $\pm 0.21$ )	
<b>OA-NER+SS</b>	45.1%( $\pm 0.22$ )	46.1%( $\pm 0.23$ )	75.6%( $\pm 0.01$ )	<b>51.7%</b> ( $\pm 0.21$ )	

Table 4.8: Various masking techniques and their performance accuracies, both in-domain and out-of-domain. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets.

overlap is also explicitly marked for all these tags with unique ids.

#### 4.7 Results

In table 4.8 we show the performance of the state-of-the-art DA model in the both fact verification datasets. Specifically, the accuracies are when the DA model was trained with each of our masking approaches, as well as with the original fully lexicalized input. Note that we show results when the same model was trained and evaluated on the same domain (e.g., Train Domain=FNC, Eval Domain=FNC), and across domains (e.g., Train Domain=FNC, Eval Domain=FEVER),.

First, we note that indeed the fully lexicalized model, which performs well in-domain, transfers very poorly to a new domain. For example, the lexicalized model trained on FEVER, gave an accuracy of 83.4% when tested on FEVER, but reduced to 48.6% when tested on FNC. This verifies our findings that the signal the model learns from unmasked text does not generalize well. Additionally, the deletion baseline performs even worse than the fully lexicalized model, which indicates that while the original text can be too domain-specific, its semantic content needs to be maintained on some level.

In terms of this work, we see that all of our masking approaches improve ac-

curacy in the cross-domain setting, by as much as 10.6% (25.8% relative), while still maintaining strong in-domain performance (dropping only a few percentage points). The best cross-domain performance is obtained using our methods which are overlap-aware, suggesting that even when content is abstracted (i.e., masked), the model benefits from explicit awareness of lexical overlap for fact verification. For example, the overlap-aware SS tagging model that was trained on FNC, gave the highest accuracy of 51.7% when tested out-of-domain, on FEVER.

## 4.8 Discussion

Next we will discuss and do some error analysis on the results from our experiments shown in table 4.8.

### 4.8.1 Delexicalization learns and preserves

One advantage of delexicalization is that it not only learns the underlying semantics, but it also preserves the existing knowledge. Consider the example given in table 4.9. Here we show the predictions made on a claim-evidence pair from the FEVER dataset<sup>7</sup> by 2 models: one which was trained on lexicalized/plain-text dataset, and the other which was trained on the same dataset which was delexicalized with the OA-NER technique described earlier in section 4.6.2. We also show the words in evidence which had the highest attention weights in both cases i.e., the words which the corresponding model thinks are the most important ones for the given task. As seen in the key evidence words column (the 6th column), in case of the model trained on the lexicalized dataset, the proper noun “Stiller” was given highest importance. This ‘**Ben Stiller effect**’<sup>8</sup> verifies our initial hypothesis that neural network based

<sup>7</sup>All these examples are from the validation partitions of the corresponding datasets.

<sup>8</sup>This phenomenon, of neural network based models relying on spurious semantic patterns in the dataset was first noticed in the spring of 2017 by one of my colleagues, (and later my collaborator in multiple papers) Dr. Sandeep Sunawal. Incidentally he noticed this weird pattern in claim-evidence pairs involving the proper noun Ben Stiller. Since it was this ‘accidental discovery’ that kick started this research (and subsequently this PhD dissertation), we fondly nicknamed this phenomenon the ‘Ben Stiller Effect’. Note that we were not the only ones (nor the first ones) to discover this phenomenon. As shown in related work, this was shown to exist by Gururangan et al. (2018a) around the same time, who named it ‘artifacts’.

models rely on patterns based on noun phrases. For example it is likely that the model was memorizing or relying on the semantic pattern that every time the proper noun “Stiller” appears the gold label is *Agree*. Hence it predicts the label also as *Agree* every time it sees this pattern, so as to match with this gold label and minimize the loss.

However, for the model that was trained on the dataset delexicalized with OANER, the key words with highest attention doesn’t include the word “Stiller”. Instead it includes PERSONc1, which is one of the named entity tags which we had replaced ‘Stiller’ with (details of this technique was described in section section 4.6.2). The mere fact that the model now gives more attention to the generic PERSONc1 instead of the proper noun ‘Stiller’ is an indication of the fact that we are able to make the model “look away” from proper nouns (and subsequently able to prevent the model from memorizing semantic patterns). This is the right intuition for a textual entailment task i.e., now the model hopefully that entailment still holds (in this particular claim-evidence example) despite different proper nouns. Thus we show that delexicalization is a step towards making the neural network based model learn the right <sup>9</sup> intuition. Note that even after the delexicalization step (which can be argued to, say, disturbs or even deletes the underlying semantic knowledge.) the model is still able to make the right prediction. Thus we show that delexicalization not only preserves the existing learning to maintain the right prediction, but also enables the model to learn the true underlying semantics needed for this given task.

#### 4.8.2 Delexicalization learns and improves

Consider another similar example given in table 4.10. In the first row, the lexicalized model predicts the label to be *agree* despite the actual label being *disagree*. In section 4.8.1 we had shown an example of how the model used the ‘Ben Stiller effect’ to correctly predict the label. However, this is an example of the model predicting the wrong label (and in turn affecting the model’s performance) due to the ‘Ben Stiller

---

<sup>9</sup>Here we use the word right, from a human intuition perspective. For example for any given claim-evidence pairs, humans are able to easily arrive at the entailment decision, agnostic to the named entities present

effect’. Note that as seen in the ‘Key Evidence Words’ column the model still places high importance to the word ‘Stiller’.

However, as seen in the second row, the model which was trained on the delexicalized version of the dataset was able to predict the correct label (*disagree*). Further, as shown in the 6th column, when using the model trained on delexicalized data, the word ‘Stiller’ is not present in the list of top 3 words which the model had given highest attention to. Thus we show that delexicalization not only learns the underlying semantic knowledge needed for entailment, but also improves the accuracy of the predicted labels.

#### 4.8.3 Delexicalization learns and improves in cross-domain setting

We empirically showed in section 4.7 that delexicalization improves learning in the cross-domain setting. Here we will show it qualitatively. Consider the example in table 4.11. Both rows show the predictions made by 2 DA models on an example from the FNC dataset Pomerleau and Rao (2017). The first row is the prediction by the model which was trained on the lexicalized version of the FEVER dataset. The second row is the prediction of the model which was trained on the delexicalized version of the FEVER dataset. As seen from the 5th column, the model trained on lexicalized FEVER data predicts the label of the FNC claim-evidence pair incorrectly. However, the model trained on the FEVER data delexicalized using OA-NER technique, predicts the label correctly. Further as seen in the 6th column, the words with the highest attention included ‘Stiller’, while in the second row it doesn’t. This is proof of the fact that delexicalization helps to learn context and also improves the prediction ability of the model by predicting the right label in cross-domain setting.

#### 4.8.4 Percentages

When all of the predictions were analyzed for this claim-evidence pair, there were a total of 24 instances where the proper noun ‘Bieber’ was present in the top 3 words from evidence which were given the highest attention. Further, in all these cases the label was also wrongly predicted by the model trained on lexicalized data (as *agree*).

However, out of these 24 examples, there were 15 instances where the model trained on delexicalized data got the prediction right (*agree*). So over 60% of the time we see this (disagree : agree) combination i.e., the model trained on lexicalized data predicted the wrong label *disagree* while the model trained on delexicalized data predicted the right label *agree*. Further, there were 5 more instances of (disagree: discuss) i.e., instances where the model trained on lexicalized data predicted the wrong label *disagree* while the model trained on delexicalized data predicted the label *discuss*. In total, 20 out of 24 times when the model trained on lexicalized had predicted the wrong label, it had blindly assigned the label *disagree* when it saw the proper noun ‘Bieber’ present in the evidence. Also, as shown above, in 15 of the 24 examples, the model trained on delexicalized data was able to correct the prediction from *disagree* to *agree*. All of this stands as proof that due to delexicalization the model not only was able to preserve context but also was also able to learn the correct label.

#### 4.8.5 Drop in performance with super sense tags

As seen from the results in table 4.8 the model trained on FNC is able to get optimal performance in the FEVER dataset when using the OA-NER + SS tag masking, while the addition of super sense tags did not benefit the model that was trained on FEVER. This could be due to the fact that overall, the evidence provided in FNC is much longer than that of FEVER, and perhaps the model needs more training data to learn stable signal from the SS tags. More importantly, this also suggests that while we have demonstrated that semantic masking is clearly beneficial, it is unclear exactly what level of masking granularity should be employed for a given domain – from coarse-grained NER tags, to the more granular super sense tags, perhaps even to very fine-grained tags such as those proposed by (Ling and Weld, 2012).

#### 4.9 Conclusion

We investigated the importance placed by neural network methods for textual entailment on various lexical items. We concluded that attention weights tend to be directed towards words that are more likely to be domain specific, which considerably impacts performance in an out-of-domain setting. To mitigate this issue, we introduced several strategies for semantically masking word classes such as nouns and verbs, by generalizing them to more abstract concepts, which are more likely to have similar usage across domains. We demonstrated the utility of our approach in a cross-domain evaluation of textual entailment for fact verification. Our approach outperforms a model trained on the original, fully-lexicalized texts by over 10% accuracy when evaluated out of domain. Since our approach is implemented as a data pre-processing step, it can potentially help any neural method that learns from text and is likely to be used out of domain.

Type of Transformation	Claim	Evidence	Actual Label	Predicted Label	Key Evidence Words
Lex	DodgeBall : A True Underdog Story has Vince Vaughn and Ben Stiller in starring roles .	DodgeBall : A True Underdog Story is a 2004 American sports comedy film written and directed by Rawson Marshall Thurber and starring Vince Vaughn and Ben Stiller .	Agree	Agree	‘True’, ‘Stiller’, ‘Dodgeball’
OA-NER	DodgeBall : A True Underdog Story has PERSONc1 and PERSONc2 in starring roles.	DodgeBall : A True Underdog Story is a DATEe1 MISCe1 sports comedy film written and directed by PERSONe1 and starring PERSONc1 and PERSONc2.	Agree	Agree	‘True’, ‘Underdog’, ‘PERSONc1’

Table 4.9: An example of prediction on a claim-evidence pair from the FEVER dataset to show that using delexicalization learns and preserves knowledge. Each of the rows show the predictions made on the same claim-evidence pair by 2 models: one which was trained on lexicalized/plain-text dataset, and the other which was trained on the same dataset which was delexicalized with the OA-NER technique. In both cases, the models predict the labels correctly. In case of the model trained on lexicalized data, as shown in the 6th column, the word ‘Stiller’ is considered as having very high importance. However when using the model trained on delexicalized data, the word ‘Stiller’ doesn’t exist in the list of words with highest importance/attention as given by the model. This is proof that delexicalization learns and preserves knowledge.

Type of Transformation	Claim	Evidence	Actual Label	Predicted Label	Key Evidence Words
Lex	DodgeBall : A True Underdog Story was unable to feature scenes with Vince Vaughn and Ben Stiller.	DodgeBall : A True Underdog Story is a 2004 American sports comedy film written and directed by Rawson Marshall Thurber and starring Vince Vaughn and Ben Stiller .	Disagree	Agree	‘True’, ‘.’, ‘Stiller’
OA-NER	DodgeBall : A True Underdog Story was unable to feature scenes with PERSONc1 and PERSONc2.	DodgeBall : A True Underdog Story is a DATEe1 MISCe1 sports comedy film written and directed by PERSONe1 and starring PERSONc1 and PERSONc2.	Disagree	Disagree	‘True’, ‘is’, ‘Underdog’,

Table 4.10: An example of a claim-evidence pair from FEVER, with the corresponding prediction from the DA model, with and without delexicalization. In the first row, we show that the lexicalized model predicts the label to be *agree* despite the actual label being *disagree*. However, as seen in the second row, when the model was trained on delexicalized data, it was able to predict the correct label, *disagree*. Further, as shown in the 6th column, the word ‘Stiller’ is not present in the top 3 words with highest attention anymore when the model trained on delexicalized data was used for prediction. This shows that delexicalization learns the underlying semantics and also improves the prediction.

Type of Transformation	Claim	Evidence	Actual Label	Predicted Label	Key Evidence Words
Lex	Bieber ringtone scares bear, saves man 's life.	A man was saved from a vicious bear attack thanks to his Justin Bieber ringtone...	Agree	Disagree	'Bieber', 'man', 'but'
OA-NER	PERSONc1 ringtone scares bear , saves man 's life.	A man was saved from a vicious bear attack thanks to his PERSONc1 ringtone . . . .	Agree	Agree	'but', 'PERSONc1', 'man'

Table 4.11: An example of how delexicalization improves prediction in the cross domain setting. Like before, both rows show the predictions made by 2 DA models on an example from the FNC dataset Pomerleau and Rao (2017). However, the difference is that these two models were trained on the FEVER dataset. The first row is the prediction by the model which was trained on the lexicalized version of the FEVER dataset. The second row is the prediction of the model which was trained on the delexicalized version of the FEVER dataset. As seen from the 5th column, the model trained on lexicalized FEVER data predicts the label of the FNC claim-evidence pair incorrectly. However, the model trained on the FEVER data delexicalized using OA-NER technique, predicts the label correctly. Further as seen in the 6th column, the words with the highest attention included 'Stiller', while in the second row it doesn't.

## CHAPTER 5

Data and Model Distillation as a Solution for  
Domain-transferable Fact Verification

In the previous chapter we showed that several neural network models depend heavily on certain statistical nuances found in the datasets, information that transfers poorly between domains. To mitigate the dependency on such artifacts, in the previous chapter we introduced a *data distillation* (or de-lexicalization) approach, which replaces some lexical artifacts such as named entities with their type and a unique id to indicate occurrence of the same artifact in claim and evidence. While promising, the risk of this direction is discarding too much information through the de-lexicalization process. For example, replacing *China* with its named entity (NE) type (COUNTRY) in an evidence sentence discards the fact that the text is about an Asian country which might be relevant in the context.

In this chapter we propose a solution that combines data distillation with *model distillation* to reduce the risk of over de-lexicalization. In particular, we introduce a teacher-student architecture inspired from that of Tarvainen and Valpola (2017). In our architecture, the student model is trained on de-lexicalized data (to take advantage of data distillation), but is also guided by a teacher trained on the original lexicalized data (as a form of model distillation) to mitigate the possibility of discarding too much lexical information.

Specifically, here we explore the combination of data and model distillation as a strategy to improve domain transfer of fact verification methods. Note that while our training process is more costly due to the combination of the student and teacher models, the output is a single individual model (the student), which has the same runtime cost as an individual classifier. Further, our approach is classifier agnostic,

and can be coupled with any fact verification method. We also investigate the domain transfer of our method between two fact verification tasks (FNC and FEVER), where we train on one and test on the other. For these experiments we couple our method with the state of the art fact verification approach based on transformers (Vaswani et al., 2017).

## 5.1 Overview

The rest of this chapter is organized as follows. In section 5.2 we will review a few publications specifically relevant to this chapter. In 5.3 we will describe the data distillation techniques used in this chapter, in addition to those from chapter 3. In section 5.4 we will elaborate on the model distillation approach which is the proposed solution in this chapter. Then, in section 5.5 we describe in detail the models used in our experiments. Section 5.7 shows the results of our experiments. We do an error analysis of these results in section 5.8 and 5.9 concludes this chapter.

## 5.2 Related Work

Before we explain the experimental setup used in this chapter, we would like to review a few related works from literature, especially the one on mean-teacher (Tarvainen and Valpola, 2017), on which the experimental setup in this chapter and chapter 6 is based on.

Mean-teacher architecture is built on the teacher-student architecture, which was first introduced in (Rasmus et al., 2015). This work suggested gamma networks as an enhancement to architectures that use noise as a regularization technique. In a teacher-student architecture the same model is used as a *teacher* and a *student*. However, the difference is that while student is a classic neural network which learns via back propagation, teacher only generates targets/predictions, which are in-turn used by both student and teacher for learning. The intuition behind gamma networks is that each data point is evaluated with and without noise (in student and teacher models respectively) and then a consistency loss is applied between the pre-

dictions of both. This multiple perspective connected via consistency loss creates a pathway using which optimized solution is arrived at. However the disadvantage of gamma networks is that since the teacher model itself creates targets, they may or may not be accurate. If the produced targets are given too much weight, the cost of inconsistency surpasses the cost of misclassification, preventing new information from being learned. In effect, the model suffers from confirmation bias, which may be reduced by increasing the target quality.

Temporal ensembling (Laine and Aila, 2016), which retains an exponential moving average (EMA) prediction for each of the training instances, was suggested as a solution to this problem. It was shown in (Laine and Aila, 2016) that temporal ensembling can enhance the model further since all the EMA predictions of the examples in that mini-batch are updated based on the new predictions at each training step. As a result, each example’s EMA prediction is made up of an ensemble of the model’s current version and previous versions that assessed the same example. The quality of the predictions improves as a result of this ensembling, and utilizing them as teacher predictions, enhances the results. However, the short coming of this method is that since each target is only updated once each epoch, the learnt data is integrated into the training process at a relatively slow rate. The longer the updates span, the larger the dataset. Also it is unclear how Temporal Ensembling can be employed at all in the case of on-line learning.

To solve this problem (Tarvainen and Valpola, 2017) introduced the **mean-teacher model**. The architecture can be found in figure 5.1.<sup>1</sup> Here the intuition is to carefully select the student model rather than blindly duplicating the teacher model. This aims to create a better student model from the teacher model. The assumption here is that a model’s softmax output rarely provides accurate predictions outside of training data. To solve this the authors suggest adding noise to the model during inference. As shown in (Gal and Ghahramani, 2016) a noisy teacher can produce more accurate goals. Specifically the authors in (Tarvainen and Valpola, 2017) propose averaging model weights instead of predictions as done

---

<sup>1</sup>Direct excerpt from the original work

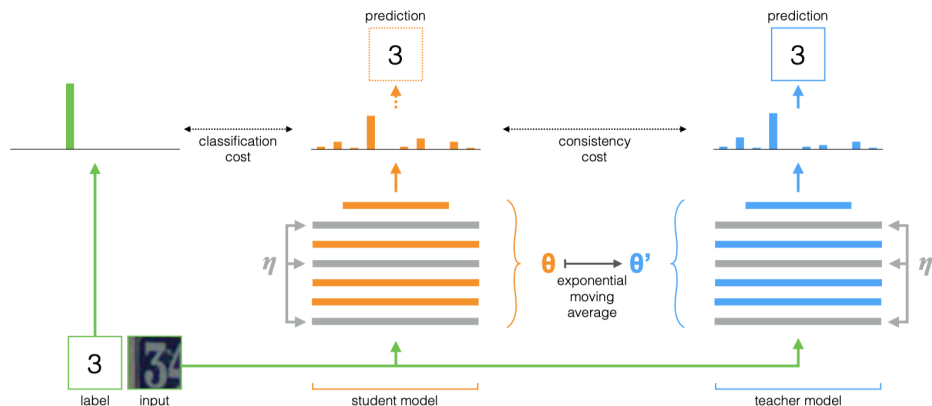


Figure 5.1: The mean teacher architecture. The figure depicts a training batch with a single labeled example. Both the student and the teacher model evaluate the input applying noise within their computation. The softmax output of the student model is compared with the one-hot label using classification cost and with the teacher output using consistency cost. After the weights of the student model have been updated with gradient descent, the teacher model weights are updated as an exponential moving average of the student weights. Both model outputs can be used for prediction, but at the end of the training the teacher prediction is more likely to be correct. A training step with an unlabeled example would be similar, except no classification cost would be applied.

in (Laine and Aila, 2016). Since the teacher model is an average of consecutive student models, this method is called the **mean-teacher** method. Further, using the final weights directly produces a more accurate model than averaging model weights throughout training steps. This can be used during training to create more accurate targets. Note that, unlike previous architectures, where the teacher model shares weights with the student model, the teacher model now uses the student model’s EMA weights. Instead of every epoch, it now aggregates data after each step. Furthermore, the target model has superior intermediate representations since the weight averages enhance all layer outputs, not just the top output.

The model setup used in the experiments in this chapter is inspired from the aforementioned teacher-student architecture and the mean-teacher architecture. This will be described in detail in section 5.4. Note that both these architectures

(teacher-student architecture and the mean-teacher architecture) were proposed for problems in computer vision. To the best of our knowledge this is the first time this architecture is being used for the recognizing textual entailment problem in natural language processing, especially in the fact verification domain. Next we will describe in detail various components of the experimental setup used in this chapter.

<b>person</b>	doctor	<b>organization</b>	terrorist_organization
actor	engineer	airline	government_agency
architect	monarch	company	government
artist	musician	educational_institution	political_party
athlete	politician	fraternity_sorority	educational_department
author	religious_leader	sports_league	military
coach	soldier	sports_team	news_agency
director	terrorist		
<b>location</b>	body_of_water	<b>product</b>	camera
city	island	engine	mobile_phone
country	mountain	airplane	computer
county	glacier	car	software
province	astral_body	ship	game
railway	cemetery	spacecraft	instrument
road	park	train	weapon
bridge			
			<b>art</b> written_work
			film newspaper
			play music
			<b>event</b> military_conflict
			attack natural_disaster
			election sports_event
			protest terrorist_attack
<b>building</b>	time	chemical_thing	website
airport	color	biological_thing	broadcast_network
dam	award	medical_treatment	broadcast_program
hospital	educational_degree	disease	tv_channel
hotel	title	symptom	currency
library	law	drug	stock_exchange
power_station	ethnicity	body_part	algorithm
restaurant	language	living_thing	programming_language
sports_facility	religion	animal	transit_system
theater	god	food	transit_line

Figure 5.2: 112 tags used in FIGER. The bold-faced tag is a rough summary of each box. The box at the bottom right corner contains mixed tags that are hard to be categorized.

### 5.3 Data distillation

In the previous chapter we demonstrated that neural network based models are most prone to over-fitting on named entities (NE) when training on fact verification datasets. There we demonstrated various masking/de-lexicalization methods as a

<b>Input Sentence</b>	<b>Named Entities</b>	<b>Named Entity Types</b>
It was won by Ottawa Senators, coached by Dave Gill.	Ottawa Senators, Dave Gill	Ottawa Senators: Organization, Sports_Team Dave Gill : Person, Coach

Table 5.1: An example of identifying and labeling several lexical expressions with FIGER (Ling and Weld, 2012) in a given sentence.

solution for this problem, including ones where we replace named entities with their type (and a unique id). However, after those experiments we observed that more fine-grained NE types yield better models. Hence, in addition to the named entity recognizer (NER) tools mentioned in section 4.6.1, in this chapter we also use the FIGER named entity recognizer (Ling and Weld, 2012) to detect and replace named entities with their most specific label returned by the NER. Note that, we also process the text with the CoreNLP NER (Manning et al., 2014) to de-lexicalize additional NER classes not covered by FIGER. We include in this list mentions of date, time, money, number, and ordinal. Next we will describe in detail about this tool and the named entity tags it provides.

### 5.3.1 FIGER

Traditional NER systems including CoreNLP (Manning et al., 2014) employ a sequence model, generally a linear-chain Conditional Random Field (CRF). Each token in a sequence model contains a corresponding hidden variable that indicates its type label. Consecutive tokens with the same type label are handled as if they were all mentioned at the same time. The hidden variables' state space is proportional to the size of the type set in this case. However, allowing a segment to have several labels expands the state space enormously. When the tag set expands to over a hundred tags, this becomes computationally infeasible. To solve this problem a pipeline based approach was proposed in (Ling and Weld, 2012). In this work the authors propose a fine-grained set of 112 tags for entity recognition generated from

	Claim	Evidence
Plain text	Mark Zuckerberg made the Forbes list of The World’s Most Powerful People	In December 2016, Zuckerberg was ranked 10th on Forbes list of The World’s Most Powerful People.
Distilled text	personC1 made the Forbes list of written_workC1’s Most Powerful People .	In December 2016, personC1 was ranked 10th on Forbes list of written_workC1 ’s Most Powerful People.

Table 5.2: An example of a claim-evidence pair from the FEVER dataset (Thorne et al., 2018b), de-lexicalized using fine-grained tags from FIGER (Ling and Weld, 2012).

Freebase. List of these tags can be found in figure 5.2.<sup>2</sup> Further, they formulate the tagging problem as a multi-class, multi-label classification problem, then describe an unsupervised training data collection technique, and implement a fine grained entity recognizer that makes use of the aforementioned fine-grained tags. In the proposed pipeline, models for segmentation and tagging are trained offline. Table 5.1 shows example outputs/tags created by FIGER.

### 5.3.2 De-lexicalization with FIGER

In this chapter we de-lexicalize the claim-evidence pairs using FIGER, in addition to the named entity recognizers mentioned in chapter 4. Similar to the de-lexicalization techniques mentioned in chapter 4, we also align the named entities between the claim and the evidence. That is, any named entity that appears first in the claim is assigned an id postfixed with  $\#Cn$ ; if an entity mention appears only in evidence then it is postfixed with  $\#En$ , where C indicates that the entity appeared first in the claim, E indicates that the entity first appeared in the evidence, and  $n$  indicates the  $n_{\text{th}}$  observed entity. Table 6.1 shows an example output for this data distillation process.

<sup>2</sup>Direct excerpt from the original work.

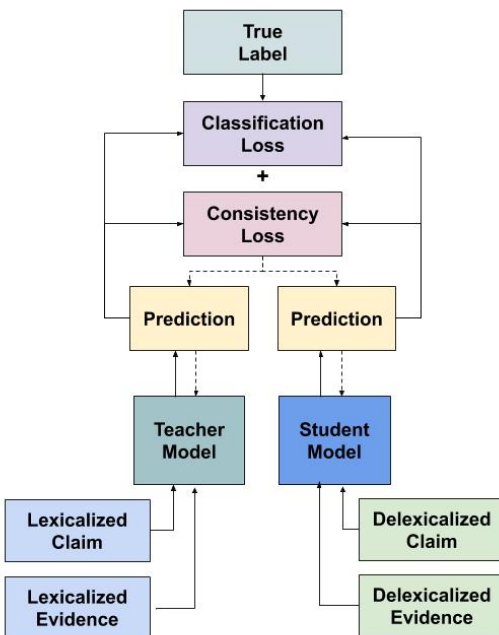


Figure 5.3: The teacher-student architecture for model distillation.

#### 5.4 Model distillation

We propose a model distillation strategy to mitigate the risk of overly aggressive data distillation. In particular, we introduce an architecture based on the teacher-student architecture Hinton et al. (2015); Tarvainen and Valpola (2017); Laine and Aila (2016); Sajjadi et al. (2016) to solve this problem. This architecture is shown in figure 5.3). In this architecture the teacher is trained on the original, lexicalized data, and the student is trained on the data de-lexicalized with the de-lexicalization approaches described before.

The intuition behind our model distillation approach is that the proposed teacher model will “pull” the student model towards the original underlying semantics, which are partially obscured to the student due to the de-lexicalization of its training data. More formally, this is captured through a consistency loss that minimizes the difference in predicted label distributions between the student and the teacher. The consistency loss is implemented as a mean squared error between the label scores

predicted by the student and the teacher. Additionally, both the student and the teacher components include a regular classification loss on their respective data, which is implemented using cross entropy. This encourages both the student and the teacher to learn as much as possible from their own views of the data.

Note that despite being inspired from them, our architecture has a few significant differences than the classic teacher-student architecture (Rasmus et al., 2015) and mean-teacher architecture (Tarvainen and Valpola, 2017). They are as described below:

1. Back propagation in teacher: In the architecture described in (Tarvainen and Valpola, 2017), the teacher model does not have any active learning i.e., unlike classic neural networks, it doesn't learn through a back propagation process for minimizing the losses between predicted and gold labels. In our architecture the teacher also is an active learner. As shown in figure 5.3 the teacher model in our architecture actively predicts labels, and the difference between this prediction and gold label is back propagated, like in classic neural network architectures. This is essential for this task because our goal in this architecture is a teacher model that learns simultaneously with the student.
2. Noise: In (Tarvainen and Valpola, 2017), both the student and the teacher model evaluate the input by applying noise within their computation. However, in our architecture, there is no explicit noise that is added. Instead each of the teacher and student models see a different version of the same dataset. While the teacher model sees the plain-text version of the data, the student model sees a de-lexicalized version of the same data (de-lexicalized using one of the aforementioned de-lexicalization techniques). Note that while it can be argued that de-lexicalization can be considered as a form of noise, we are considering the classic definition of noise here i.e., perturbations sampled from a known distribution (e.g., Gaussian) and added to the input data to modify it.
3. Embeddings: The input data in (Tarvainen and Valpola, 2017) is raw pixel data while we use word embeddings. Granted, that the problem in (Tarvainen

and Valpola, 2017) is from the computer vision domain, and the problem we are tackling is in the domain of natural language processing. However, one difference is that in our architectures the embeddings are updated during training. Hence, the learning is continuously reflected and modified in the input space. More importantly, while the initial embeddings of the tokens in input are initialized based on those from GLOVE (Pennington et al., 2014), the tokens used in de-lexicalization is created meaningfully. To elaborate, we create pseudo-pretrained embeddings for the de-lexicalization tokens by adding a small amount of random Gaussian noise (mean 0 and variance of 0.1) to pre-trained embeddings (Pennington et al., 2014) of the root word corresponding to the category. For example in the table 6.1 when the named entity ‘Mark Zuckerberg’ is replaced with `personC1`, the initial embedding of `personC1` is created by adding a small amount of random Gaussian noise to pre-trained embedding of the root word corresponding to the category (e.g., `person`). Thus the embeddings of all the sub-tags, while being unique, are close to that of the root word.

## 5.5 Models

In this chapter all our experiments are done with a state-of-the-art method for fact verification, transformers (Vaswani et al., 2017), which has achieved state-of-the-art results not only in the task of fact verification but in several other NLP tasks. Specifically, we use the PyTorch implementation of BERT Devlin et al. (2019) from huggingface (Wolf et al., 2019). We experimented with several pre-trained BERT-base models and found that the one which gave the highest performance was the BERT-cased model when used with a sequence length of 128. Further, to distinguish the vocabulary of the de-lexicalized data from the lexicalized data we augment the base vocabulary of BERT with tokens specific to the de-lexicalized data. For example, as mentioned before, during de-lexicalization we use `personC1` to denote the first occurrence of the named entity in the claim paragraph. However, to ensure

that the BERT BasicTokenizer does split `personC1` into `person` and `C1`, we added the token “C1” to the BERT vocabulary. Tokenizers for each of the lexicalized and de-lexicalized dataset are initially created using BERT BasicTokenizer, but then use the aforementioned vocabulary created for the specific data type.

Train Domain	Configuration			
	FEVER	FEVER	FNC	FNC
Eval Domain	FEVER	FNC	FNC	FEVER
BERT Lex	94.5% ( $\pm 0.13$ )	68.9% ( $\pm 0.03$ )	96.9% ( $\pm 0.02$ )	73.1% ( $\pm 0.07$ )
BERT Delex (OA-NER)	82.1% ( $\pm 0.2$ )	53.9% ( $\pm 0.09$ )	65.5% ( $\pm 0.04$ )	46.4% ( $\pm 0.1$ )
BERT Delex (OA-NER + SS)	75.6% ( $\pm 0.08$ )	46.1% ( $\pm 0.11$ )	45.1% ( $\pm 0.09$ )	51.7% ( $\pm 0.08$ )
BERT Delex (FIGER)	91.7% ( $\pm 0.01$ )	54.7% ( $\pm 0.08$ )	96.2% ( $\pm 0.07$ )	62.9% ( $\pm 0.12$ )
BERT TS (FIGER)	89.2% ( $\pm 0.11$ )	<b>73.4%*</b> ( $\pm 0.02$ )	98.9% ( $\pm 0.11$ )	<b>74.8%*</b> ( $\pm 0.12$ )

Table 5.3: In-domain and cross-domain accuracies for various methods. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets. “BERT Lex” is the stand alone model trained on the original lexicalized data; “BERT Delex” is the standalone model trained on de-lexicalized data. OA-NER de-lexicalizes the data using the Overlap Aware Named Entity Recognizer; SS uses Super Sense tags — two de-lexicalization techniques mentioned in Suntwal et al. (2019a). FIGER de-lexicalizes the data using a fine-grained named entity recognizer (Ling and Weld, 2012). ; and “BERT TS” denotes the student in the proposed teacher-student architecture. \* indicates that the corresponding result is significantly better than its baseline (“BERT lex” in the same column), under a bootstrap resampling test with 1,000 samples, and  $p$ -value  $< 0.035$ .

## 5.6 Experiments

In all the experiments, the performance of the underlying model on the respective lexicalized data is considered as the baseline. For example when training a teacher-student model on FEVER, the baseline is the model that was trained using the original text of the FEVER dataset. In the baseline model, we use the default hyper parameters set in the huggingface repository (Wolf et al., 2019). We focus our analysis on cross-domain evaluation, i.e., we train all models on one dataset (e.g., FEVER) and evaluate their accuracy on the other dataset (e.g., FNC).

## 5.7 Results

Table 5.3 summarizes the results of our experiments with various models tested in-domain and cross-domain. All scores reported are averaged across three random seeds. We use ‘BERT Lex’ as the baseline model which is the stand alone model trained on the original lexicalized data. ‘BERT Delex’ denotes the standalone models trained on de-lexicalized data, along with the corresponding delexicalization techniques used. OA-NER uses the Overlap Aware Named Entity Recognizer for delexicalization of data and SS uses Super Sense tags (Suntwal et al., 2019a). FIGER delexicalizes the data using a fine-grained named entity recognizer (Ling and Weld, 2012). ‘BERT TS’ denotes the student in the proposed teacher-student architecture. Since the delexicalization used by the best performing ‘BERT Delex’ models in the cross-domain setting was FIGER, we chose it as the preferred delexicalization technique for this student.

Note that the lexicalized models, which perform well in-domain, tend to transfer poorly to a new domain. For example, the BERT model trained on lexicalized FEVER data, gave an accuracy of 94.5% when tested on FEVER, but reduced to 68.3% when tested on FNC. This verifies our findings that the signal the model learns from unmasked text does not generalize well.

In contrast, in all our experiments, the student models trained under the teacher-student architecture outperform the other models trained using lexicalized data, in a cross-domain setting. For example, the student model of the teacher-student architecture trained on FEVER, gave an accuracy of 89.2% when tested on FEVER and an accuracy of 73.4% when tested on FNC. Similarly in the other direction, when the same model was trained on FNC, it gave an accuracy of 98.9% when tested on FNC, and an accuracy of 74.8% when tested on FEVER. Note that in both the directions the accuracy of the student model of the teacher-student architecture surpasses the corresponding accuracy of the model trained on lexicalized data in a cross-domain setting. These experiments were repeated under a bootstrap resampling test with 1,000 samples, and  $p$ -value  $< 0.035$  to ensure statistical significance.

## 5.8 Error Analysis

We believe that the improved performance of the student model in the TS architecture is due to the fact that the TS architecture provides additional information over the ground labels. The key addition of our TS approach is that the de-lexicalized student learns to mimic the label probability distributions of the teacher through the consistency loss. As discussed earlier, we conjecture that this pulls the student model closer to the teacher. Another possible interpretation is that the model distillation has a regularization effect since the consistency loss essentially averages the behavior of both models.

Importantly, our results indicate that too much delexicalization risks discarding useful information. We believe this is why the standalone de-lexicalized model performs worse out of domain, and why the TS de-lexicalized student performs better. Understanding *how much* delexicalization to apply given a task opens up interesting avenues for future research. Nevertheless, overall this chapter demonstrates that data distillation and model distillation can be combined as a strategy to improve domain transfer of fact verification methods.

Lex	TS Student
Apple	year
year	said
Rivers	<b>country</b>
said	<b>person</b>
Islamic	according
State	<b>organization</b>
according	news
says	Islamic
Watch	<b>engineer</b>
report	<b>actor</b>

Table 5.4: Top 10 tokens with the highest attention weights by each of the trained models. ‘Lex’ is the stand alone model trained on the original lexicalized data and ‘TS Student’ denotes the student in the proposed teacher-student architecture.

Lastly, we also inspected the word-level attention weights (Bahdanau et al., 2014) to further understand what these models are learning. Specifically, we analyze

the weights assigned by the last attention head in the last layer of the respective transformer models. Table 5.4 shows the tokens that were assigned highest weights by the model trained on lexicalized data and the teacher-student model.<sup>3</sup> It can be seen that the tokens that were given the highest weights by the model trained on lexicalized data contain more named entities (e.g., *Apple*, *State*). This suggests potential overfitting, since the specific named entities should not be relevant for the fact verification task.

On the other hand, the tokens that were given the highest weights by the teacher-student model contain more generic named entity labels (e.g., country, person). Also we found that out of all the attention weights assigned by the model trained on lexicalized data, 15.0% were given to named entities. Further, in the TS student model only 7.4% was assigned to named entity labels. These findings demonstrate that by using the data distillation and model distillation techniques we are able to reduce the importance that models place on lexical artifacts. This not only helps them achieve accuracies at par with their counterparts trained on plain text data in an in-domain setting, but also outperform them in a cross-domain setting.

## 5.9 Conclusion

We present a new strategy to improve domain transfer of fact verification methods, which combines data distillation and model distillation. We show that the performance of existing state-of-the-art models degrades significantly on a cross-domain setting, hence motivating the necessity of robust data distillation techniques such as delexicalization to minimize overfitting on lexical artifacts. We further combine delexicalization with a teacher-student architecture as a form of model distillation to reduce the risk of over-delexicalization. We hope that this solution will encourage the development of architectures capable of reducing the dependency of models on lexical artifacts in an effort to learn domain transferable knowledge in the task of fact verification.

---

<sup>3</sup>Stop words and other BERT specific tokens like [SEP], [CLS], [PAD], etc., are removed from this list.

## CHAPTER 6

Group Learning: A Collective Knowledge Distillation Solution for Domain  
Transfer in Fact Verification

In the previous chapters, we showed that there are limitations to state of the art neural network methods in transferring data across domains, caused in part by their over-fitting on statistical and lexical nuances (or artifacts) specific to a dataset (Gururangan et al., 2018a). As shown in chapter 4, a key solution to solving this problem is to not let these models rely on such dataset artifacts, and instead encode the true underlying semantics of the dataset. For this we created a data distillation (or de-lexicalization) approach, which replaces some lexical artifacts such as named entities with their type and a unique id to indicate occurrence of the same artifact in claim and evidence. Further in chapter 5 we show that combining data distillation with a model distillation approach controls over-de-lexicalization.

However, a key unresolved issue still is *how much* de-lexicalization to apply. As indicated in previous chapters, de-lexicalization reduces over-fitting. But too much de-lexicalization may discard critical information. For example, replacing *India* with its NE label, say **LOCATION**, may remove contextual information about the country that is necessary for the correct classification of the claim-evidence pair. In this chapter we propose a solution for this problem. Inspired by teacher-student from previous chapter, we propose a novel architecture that combines data distillation with model distillation to improve cross-domain performance of neural networks. In particular, our approach relies on multiple students that have access to different de-lexicalized views of the data, but are encouraged to learn from each other through pair-wise consistency losses. We call our approach *Group Learning (GL)*. Once training completes, the student with the best performance is kept for evaluation purposes. Because we rely on a single model at evaluation time, our approach has

the same evaluation run time cost as a single classifier. Note that our method can be seen as an inverse of an ensemble strategy, which trains individual models separately but applies them jointly. However, the difference is that GL trains all the models together and picks one of the best models from the trained models for predictions. Further, GL scales better at inference time due to its reliance on a single model at that stage. Here we implement a GL architecture for fact verification using BERT (Devlin et al., 2019) as the classifier, and multiple de-lexicalized views of the data using FIGER and CoreNLP NER systems, and let GL select the optimal de-lexicalization approach.. We then evaluate the domain transfer of the proposed method using two fact verification datasets as before: FEVER and FNC. Our results show that our method achieves a cross-domain accuracy higher than that produced by previous two methods. Importantly, our approach chooses different students in each direction, highlighting different properties of the respective training datasets.

Like in the previous chapters, the approach in this chapter also involves data distillation/de-lexicalization as the first step and model distillation as the second. However, in this chapter we experiment with a few more new de-lexicalization techniques availing more granularity. Further, we use an extended version of the teacher-student architecture from chapter 5 to decide on the right level of de-lexicalization. Next we will elaborate on these two steps and how they are different in this chapter.

## 6.1 Data Distillation

Based on the findings of previous chapters that named entities (NEs) are most likely to over-fit in a fact verification task, in this chapter also we de-lexicalize our data by replacing NEs with their semantic classes (and a unique id). To detect and replace named entities with their most specific label returned by the Named Entity Recognizer (NER) we use the same solutions from section 5.3 of chapter 4. However, to explore some more levels of granularity, in this chapter we experiment with two more de-lexicalization techniques based on the FIGER-NER (Ling and Weld, 2012) called FIGER-abstract and FIGER-specific. In FIGER-abstract, we replace the named entities with the abstract classes returned by the FIGER NER,

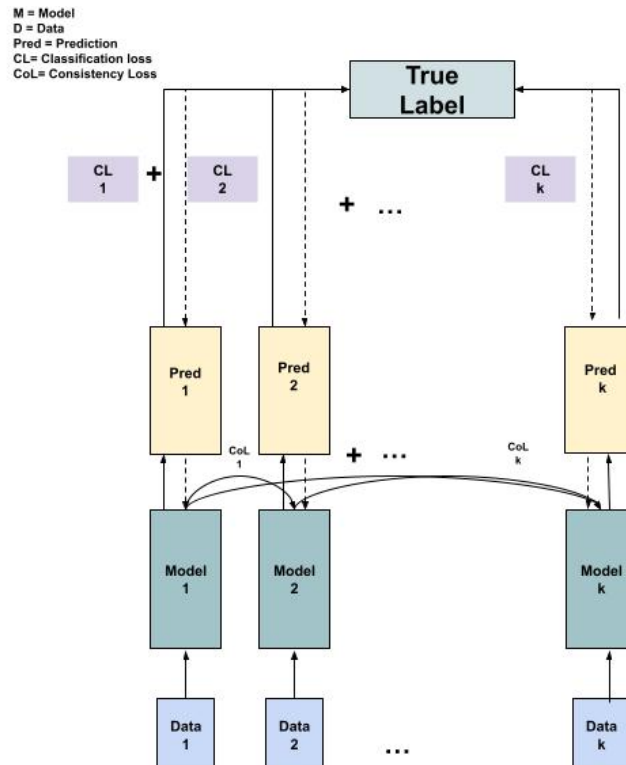


Figure 6.1: The multiple-student architecture for knowledge distillation.

(e.g., LOCATION for *Los Angeles*). In FIGER-specific we use more specific classes returned by the FIGER NER, (e.g., CITY for *Los Angeles*).

## 6.2 Model Distillation

To mitigate the risk of distilling the data at the incorrect granularity (overly aggressive or too conservative), we propose a combined distillation strategy. Specifically, we introduce a *Group Learning* architecture (shown in Figure 6.1), inspired from the teacher-student paradigm (Hinton et al., 2015; Tarvainen and Valpola, 2017; Laine and Aila, 2016; Sajjadi et al., 2016). In this architecture each student method is trained on two techniques:

(a) Different versions of the same dataset, each de-lexicalized differently by using different data distillation techniques mentioned above. An example of each is shown

in Figure 6.1.

(b) The distributions of predictions of other models.

This combined methodology of knowledge distillation encourages students to learn as much as possible from their own views of the data, while jointly learning with other students. Training together on the soft labels (distribution of predictions) of other student methods acts as a form of regularization between all methods. More formally, each student component includes a regular classification loss (implemented using cross entropy) on their respective data. Additionally, they each have a consistency loss between all other methods that minimizes the difference in predicted label distributions between them.

The intuition behind our approach is that by providing multiple data distillation options to choose from we encourage the student methods to ‘pull’ towards each other and the original underlying semantics. The part of semantic knowledge that is obscured from a student method due to the particular de-lexicalization technique used in the dataset version it sees, is instead learned in its effort to perform on par with other methods. Thus, similar to a classroom environment where the students learn from both, known labels (e.g., a textbook) and by helping another student learn, each student is able to thus choose the right amount of granularity needed to enhance its own understanding.

## 6.3 Models and Data

### 6.3.1 Models

Similar to what was used in chapter 5, BERT (Devlin et al., 2019) is the pre-trained model used in our experiments in this chapter also. However, in this chapter the difference is that we experiment with two variants of BERT, BERT-Base, and Mini BERT (Turc et al., 2019) (a light-weight version of BERT), both from the Hugging Face repository (Wolf et al., 2019). The rest of the details of the models remain same as described in section 5.5, including how the vocabulary of BERT was modified to include the custom de-lexicalization tokens.

### 6.3.2 Data

The data remains same as in previous chapters. Hence we refer you to section 4.4.1 for details on the datasets used for cross domain evaluation.

## 6.4 Experiments

In all the experiments, the performance of the underlying method on the respective original, lexicalized data is considered as the baseline. In the baseline model, we use the default hyper parameters from Hugging Face. We focus our analysis on cross-domain evaluation, i.e., we train all methods on one dataset (e.g., FEVER) and evaluate their accuracy on the other dataset (e.g., FNC). At the end of training, the best student model from the list of all the trained models is saved to be used for evaluation.

## 6.5 Results

Table 6.2 summarizes our results. We focus on two sets of experiments using each training method and setting: in-domain (columns 2 and 4) and cross-domain (columns 3 and 5). Although all models perform well (83.8%–99.5%) in-domain, they transfer poorly when evaluated cross-domain where up to 35% drop in performance is observed. This verifies our findings that the signal the model learns from un-masked text does not generalize well between domains. On the other hand, we observe a marginal in-domain drop in performance for the student models trained on the GL architecture (e.g., 9.5% in the FEVER/FEVER setting for BERT-Base Cased) compared to their lexical counter parts. This indicates that GL models retain most signal from lexical data. Importantly, the GL models perform considerably better than their corresponding lexical versions across domain (e.g., up to 20.3% improvement in the FNC/FEVER setting for Mini-BERT). This demonstrates that data distillation and model distillation can be successfully combined as a strategy to improve domain transfer of fact verification methods.

## 6.6 Discussion

In this section we will do some error analysis and discuss some interesting discoveries that were made during experiments in this chapter.

### 6.6.1 Which de-lexicalization technique is the best.

While analyzing the selection made by the GL framework for various random seeds, we observed that when trained on FEVER and tested on FNC, GL selects the lexicalized student, while in the other cross-domain direction the common choice is the student de-lexicalized with OA-NER. We hypothesize this happens for two reasons. First, the training data in the FNC dataset is smaller (40,904 data points) when compared to that of the FEVER dataset (119,197 training data points), so it is more prone to over-fitting in the original, lexicalized form. Second, since the FNC dataset is derived from real-world news articles, the number of evidence sentences in the FNC are higher than the sentences in FEVER. This means that de-lexicalized sentences in FNC preserve enough lexical signal for training, even in their de-lexicalized forms. The opposite observations hold in the other direction (FEVER to FNC), which caused the lexicalized students to perform better. Also do note that even though we use only 4 student methods in our experiments to train with each other, this can be extended to any number of methods. However, the right number of student models (and their corresponding de-lexicalized datasets) for a given task needs to be empirically determined.

### 6.6.2 Higher attention to tokens across claim-evidence boundaries.

It has been shown in literature (Clark et al., 2019) that when pre-trained BERT is used for fine-tuning domain-specific data, highest attention weights <sup>1</sup> are given to BERT specific tokens like [CLS] (a special token added to denote the beginning of a text) and [SEP] (a special token added to denote the end of a text). To verify this

---

<sup>1</sup>For details on what attention weight is please refer to the original work on Transformer networks (Vaswani et al., 2017).

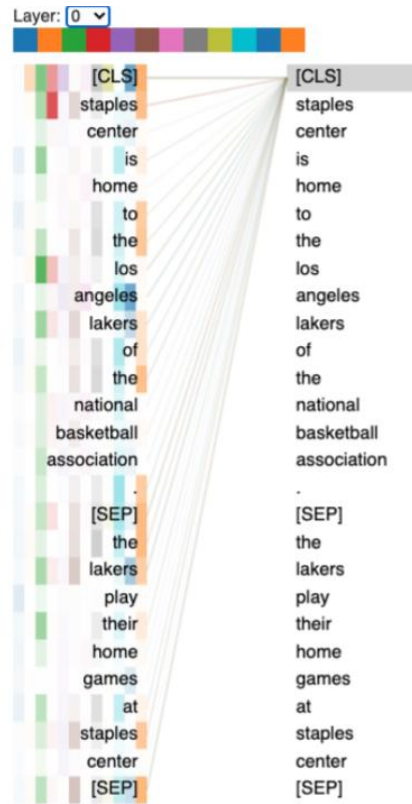


Figure 6.2: Weights assigned by the attention heads in the first layer of BERT for a given claim-evidence pair when predicting using a student model trained on plain-text data. It can be seen that despite multiple tokens available from the given claim-evidence pair, most attention weight is assigned to the BERT specific tokens like [CLS] and [SEP].

consider the attention weights plotted from our experimental setup for a given claim-evidence pair shown in Figure 6.2 and Figure 6.3. These are attention weights across the first and last layers of BERT’s attention head when using a student model trained on plain-text data. As can be seen from these examples, the attention weights in the first and final layers of BERT are all assigned to the BERT specific tokens like [CLS] and [SEP].

Further, (Kovaleva et al., 2019) recently showed that intra-sentence attention is very minimal in BERT. Intra-sentence attention, is the attention that BERT heads assign between tokens in sentences in tasks that have two distinct parts (e.g., the claim-evidence pair given in Figure 6.2 and Figure 6.3).

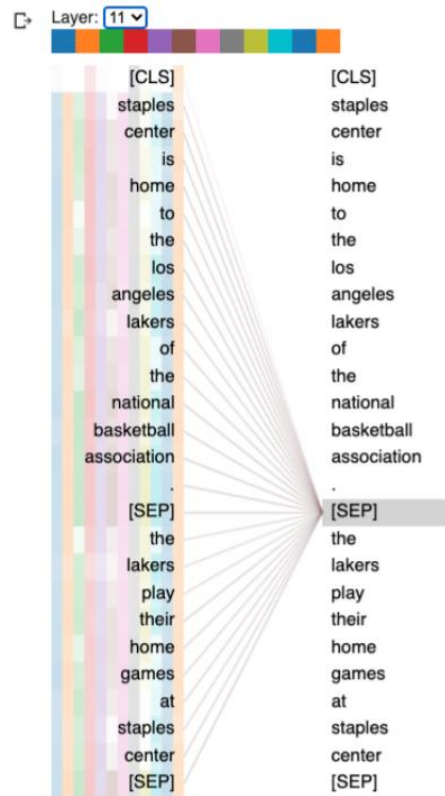


Figure 6.3: Weights assigned by the attention heads in the last layer of BERT for a given claim-evidence pair when predicting using a student model trained on plain-text data. It can be seen that despite multiple tokens available from the given claim-evidence pair, most attention weight is assigned to the BERT specific tokens like [CLS] and [SEP].

To verify if this is true in a fact verification setting, we calculated the percentage of all attention weights for all the claim-evidence pairs in the current fact verification experiment in a student model that was trained using plain-text data. As shown in Figure 6.4, for all the claims in the FEVER dataset, 62.1% of all attention weights came from tokens in the same claim text itself, while only 37.9% came from the corresponding evidence text.

The authors in (Kovaleva et al., 2019) argue that BERT thus classifying a claim-evidence pair, without placing attention on tokens across from claim (e.g., tokens from evidence text) is opposite to the human intuition on solving RTE tasks. Hence they suggest that BERT models are significantly over-parametrized. All of this cor-

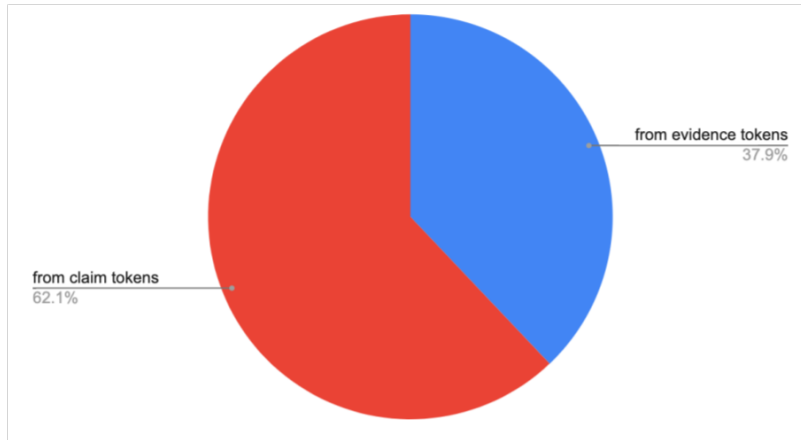


Figure 6.4: The percentage of all attention weights for all the claim-evidence pairs in the fact verification dataset using a student model that was trained using plain-text data. For all the claims in the FEVER dataset, 62.1% of all attention weights came from tokens in the same claim text itself, while only 37.9% came from the corresponding evidence text.

roborates with our initial hypothesis on over-fitting which was proposed in chapter 4.

However, after analyzing the same attention weights placed by the student model, which was trained with de-lexicalized data, showed that for a given claim text, de-lexicalization makes BERT place more attention to tokens from evidence text. For example Figure 6.5 shows the calculated the percentage of all attention weights for all the claim-evidence pairs in the current fact verification experiment in a student model that was trained using de-lexicalized version of the same data. It can be now seen that for all the claims in the FEVER dataset, only 37.5% of all attention weights come from tokens in the same claim text itself, while 62.5% come from the corresponding evidence text. This proves that using de-lexicalization techniques in a teacher-student architecture encourages BERT to place more attention weights to tokens across the claim-evidence boundary. This is more close to human intuition

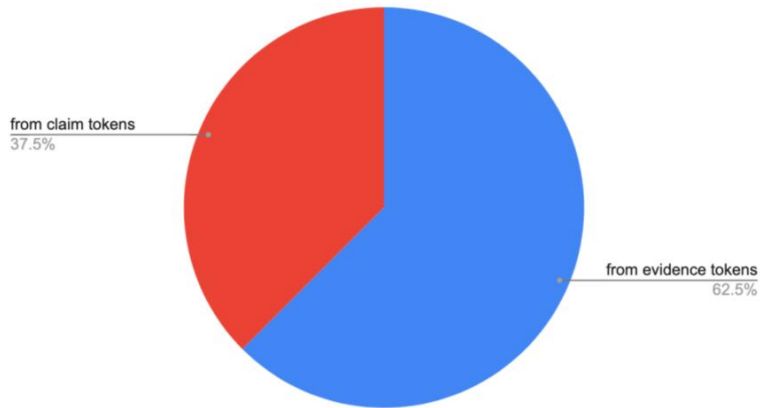


Figure 6.5: The percentage of all attention weights for all the claim-evidence pairs in the fact verification dataset using a student model that was trained using de-lexicalized data. For all the claims in the FEVER dataset, only 37.5% of all attention weights come from tokens in the same claim text itself, while 62.5% come from the corresponding evidence text.

than BERT placing attention to custom tokens (e.g., [CLS] and [SEP]) or tokens from the same claim text itself.

## 6.7 Conclusion

In the previous chapters we showed that de-lexicalization is useful in learning domain transferable knowledge. However, the level of de-lexicalization suitable for each task is unclear. In this chapter, we introduce a new architecture which provides multiple de-lexicalization choices to neural network models and encourage them to choose the most appropriate choice. We suspect this approach acts as regularization (through the consistency losses), as well as a form of data noise (because of the imperfect NERs), which has been shown to aid in knowledge distillation paradigms (Hinton et al., 2015; Tarvainen and Valpola, 2017). In conclusion, the approach in this chapter demonstrates that: (a) de-lexicalization helps model generalization, (b) the

amount of de-lexicalization to apply varies from dataset to dataset, and (c) it is possible to learn how much de-lexicalization to apply through our proposed GL architecture.

Type of Transformation	Claim	Evidence
Plain text	<i>J. R. R. Tolkien</i> created <i>Gimli</i> .	A dwarf warrior , he is the son of <i>Glóin</i> -LRB- a character from <i>Tolkien</i> 's earlier novel, <i>The Hobbit</i> -RRB-. <i>Gimli</i> is a fictional character from <i>J. R. R. Tolkien</i> 's <i>Middle-earth</i> legendarium, featured in <i>The Lord of the Rings</i> .
Using named entity recognizer from Stanford	<b>personC1</b> created <b>personC2</b> .	A dwarf warrior, he is the son of <b>personE1</b> -LRB- a character from <b>personC1</b> 's earlier novel, <i>The Hobbit</i> -RRB-. <b>personC2</b> is a fictional character from <b>personC1</b> 's <b>locationE1</b> legendarium, featured in <i>The Lord of the Rings</i> .
Using abstract named entities produced by a fine grained entity recognizer	<b>personC1</b> created <b>locationC1</b>	A dwarf warrior, he is the son of <b>personE1</b> -LRB- a character from <b>personC1</b> 's earlier novel, <i>The Hobbit</i> -RRB-. <b>locationC1</b> is a fictional character from <b>personC1</b> 's <b>written_workE1</b> legendarium, featured in <i>The Lord of the Rings</i> .
Using specific named entities produced by a fine grained entity recognizer	<b>authorC1</b> created <b>locationC1</b> .	A dwarf warrior , he is the son of <b>personE1</b> -LRB- a character from <b>authorE1</b> 's earlier novel, <i>The Hobbit</i> -RRB- . <b>locationC1</b> is a fictional character from <b>authorC1</b> 's <b>written_workE1</b> legendarium , featured in <i>The Lord of the Rings</i> .
Using specific named entities produced by a super sense entity tagger	<b>personC1</b> <b>creationC1</b> Gimli.	A dwarf warrior, he <b>stativeE1</b> the son of <b>personE1</b> -LRB- a character from <b>personC1</b> 's earlier novel, <i>The Hobbit</i> -RRB-. <b>personC2</b> <b>stativeE1</b> a fictional character from <b>personC1</b> 's <b>locationE1</b> legendarium, <b>socialE1</b> in <i>The Lord of the Rings</i> .

Table 6.1: Examples of a claim-evidence pair of sentences used in fact verification being transformed using various delexicalization techniques. The delexicalization labels range from more abstract to more specific. For example the first row shows the given claim evidence pair in its original plain text format. In the second row the same text is delexicalized by replacing some concepts with their corresponding named entities identified using the Stanford named entity recognizer (Finkel and Manning, 2009). The third row shows the same data delexicalized with an entity recognizer from AllenAI (Ling and Weld, 2012) and a set of abstract types. The fourth row uses the same entity recognizer, but with more specific, i.e., more fine-grained, entity types. For example, we now replace *Tolkien* with **author** rather than **person**.

Configuration				
Train	FEVER	FEVER	FNC	FNC
Eval	FEVER	FNC	FNC	FEVER
Mini BERT Lex	83.8% ( $\pm 0.12$ )	69.5% ( $\pm 0.64$ )	89.3% ( $\pm 0.23$ )	54.1% ( $\pm 0.02$ )
Mini BERT <b>GL</b>	83.4% ( $\pm 0.02$ )	73.6%* ( $\pm 0.15$ )	89.2% ( $\pm 0.07$ )	74.6%* ( $\pm 0.42$ )
BERT-Base Cased Lex	90.8% ( $\pm 0.03$ )	66.8% ( $\pm 0.03$ )	90.8% ( $\pm 0.03$ )	73.8% ( $\pm 0.03$ )
BERT-Base Cased <b>GL</b>	84.8% ( $\pm 0.03$ )	<b>75.7%*</b> ( $\pm 0.03$ )	87.7% ( $\pm 0.03$ )	75.1% ( $\pm 0.03$ )
BERT-Base Uncased Lex	91.5% ( $\pm 0.03$ )	64.1% ( $\pm 0.03$ )	91.5% ( $\pm 0.03$ )	76.9% ( $\pm 0.03$ )
BERT-Base Uncased <b>GL</b>	86.2% ( $\pm 0.03$ )	73.1%* ( $\pm 0.03$ )	98.2% ( $\pm 0.03$ )	<b>77.7%</b> ( $\pm 0.1$ )

Table 6.2: In-domain and cross-domain accuracies for various methods. The accuracies reported are the mean across 3 random seed experiments using the BERT model, with their corresponding standard deviations mentioned in brackets. “Lex” is the stand alone model trained on the original lexicalized data; “GL” denotes the student in the proposed multi-student ‘Group Learning’ architecture. \* indicates that the corresponding result is significantly better than its baseline (“lex” in the same column), under a bootstrap resampling test with 1,000 samples, and  $p$ -value  $< 0.05$ .

## Conclusion

In this work, we addressed the task of domain adaptation in Natural Language Processing (NLP). Domain adaptation is the ability machine learning models to transfer the knowledge learned when training in one domain, to another domain without the need of re-training in the new domain. Specifically, we focused on neural network based methods and showed how the problem of over-fitting prevents neural networks from achieving robust domain adaptation. Further, we proposed a few solutions to address this challenge. The solutions we developed were tested on a specific NLP task: fact verification. Through our solutions of data and model distillation we show that robust domain adaptation for fact verification tasks can be achieved.

The task of fact verification is determining if assertions made in written or spoken language are correct. Currently, fact verification plays an important part in fake information detection pipelines. For this purpose many automated machine learning systems have been developed which re-define and address fact verification as a task of recognizing textual entailment. However, one problem that has plagued many of these automated fact verification architectures is the inability of neural network based methods to transfer learning from one domain to another. For example, as shown in chapter 3, a neural network method trained on a fact verification dataset created using facts from the Wikipedia domain, was not able to transfer its learning when asked to do fact verification in another dataset created from news articles.

As a first step of tackling this problem we investigated the root cause of this problem. Specifically, in chapter 3 we looked at the weight given to distinct lexical terms by neural network approaches when solving fact verification problems. We concluded that attention weights are biased toward terms that are more likely to be domain specific, which has a significant influence on performance in an out-of-domain context. Thus we arrived at the conclusion that there is a necessity for

approaches to lessen this dependency of neural network methods on lexicalized to achieve robust domain adaptation.

Next, to mitigate this issue, we introduced several strategies for semantically masking word classes such as nouns and verbs in chapter 4. Specifically, we generalize these word classes to more abstract concepts, which are more likely to have similar usage across domains. For example, we explored various masking techniques, called de-lexicalization techniques, such as Deletion (where each of the named entities are deleted from the sentences), overlap aware NER (where we capture the lexical overlap between the claim and evidence sentences by assigning unique entity ids to the named entity tags) etc. Further, the aforementioned techniques are combined with other named entity tags of varying levels of granularity such as Super Sense (SS) Tags and FIGER (Fine grained Entity Recognizer). Our findings indicated that these de-lexicalization techniques deliver results in-domain (i.e., where we test in the same domain in which the model was trained) comparable to state-of-the-art algorithms trained on lexicalized data. We further show that systems trained on de-lexicalized data perform significantly better out-of-domain, confirming the relevance of de-lexicalization for domain transferability. Since our approach is implemented as a data pre-processing step, it can potentially help any neural method that learns from text and is likely to be used out of domain. To provide this methodology as a means of carrying out more relevant machine learning experiments, we also released de-lexicalized versions of the several fact verification datasets.

However, a problem with the afore-mentioned data-distillation solutions is finding how much to de-lexicalize. We solve this problem, in chapter 5, by combining data distillation with model distillation into Knowledge Distillation. Specifically, we combine the afore-mentioned delexicalization techniques with a teacher-student architecture as a form of model distillation to reduce the risk of over-delexicalization. In our architecture, the student model is trained on de-lexicalized data (to take advantage of data distillation), but is also guided by a teacher trained on the original lexicalized data (as a form of model distillation) to mitigate the possibility of discarding too much lexical information. We showed that the student models trained

under the teacher-student architecture outperform the other models trained using lexicalized data, in a cross-domain setting. Further, we showed that our approach is classifier agnostic, since it can be coupled with any fact verification method.

The problem with aforementioned knowledge distillation architecture is that the level of de-lexicalization suitable for each task is unclear. In chapter 6, we introduce a new architecture, called group learning architecture, which provides multiple de-lexicalization choices to neural network models and encourage them to choose the most appropriate choice through pair-wise consistency losses. We showed that the group learning based models perform considerably better than their corresponding lexical counterparts, with upto 20.36% improvement in the cross-domain setting.

Beyond the specific techniques we have introduced here, the idea of having neural network models that can transfer learning from one domain to another, holds promise as a proving ground for developing novel machine learning methods in the future. One venue of possible future work is to explore methods for knowledge and model distillation dynamically. For example, the most appropriate level of delexicalization for each concept based on its surrounding context can be learned using reinforcement learning. In this method, each concept can be considered to exist in multiple delexicalized states and the optimization approach can be to improve cross-domain performance by rewarding the model for choosing the state for each concept that performs best in the cross-domain setting. This will not only reduce dependency on domain-specific data but also provide domain agnostic information to build robust graphs that can be queried. Another possible future work is to address another challenge faced by current fact verification's systems in their inability to capture sufficient background knowledge, which in turn leads to unscalable fact verification approaches. To address these limitations, our work can be used to build novel and robust methods for fact verification that aggregate information across multiple statements and reach global conclusions from this aggregation. For example, to capture all the real-world knowledge, relevant facts can be mined and included into a largely domain-agnostic graph, and then novel graph algorithms can be used to aggregate this knowledge.

To conclude, we showed that the methods we proposed in this dissertation helps create neural network architectures that provide robust domain adaptation for fact verification tasks. The advantage of our techniques is that the neural network methods created using our techniques need to be only trained once but can be reused multiple times for prediction without having to re-train them. This will reduce the heavy overhead of repeated training cycles, and the subsequent carbon footprint that generates. Along with making domain adaptation more robust, various techniques introduced in this dissertation also provides ground work to future architectures and solutions which can potentially accomplish information aggregation enhancing the ability of fact verification solutions.

## APPENDIX

Some examples of claim and evidence sentences from FNC dataset (Pomerleau and Rao, 2017)

## Example 1

**Claim:** Pope Francis turns out not to have made pets in heaven comment.  
**Evidence:** NEW YORK — Pope Francis has given hope to gays , unmarried couples and advocates of the Big Bang theory . Now , he has endeared himself to dog lovers , animal-rights activists and vegans . Trying to console a little boy whose dog had died , Pope Francis told him in a recent public appearance on St Peter ’ s Square that “ paradise is open to all of God ’ s creatures ” . While it is unclear whether the Pope ’ s remarks helped soothe the child , they were welcomed by groups such as the Humane Society and People for the Ethical Treatment of Animals ( PETA ) . In his relatively short tenure as leader of the world ’ s one billion Roman Catholics since taking over from Benedict XVI , Pope Francis , 77 , has repeatedly caused a stir among conservatives in the church . He has suggested more lenient positions than his predecessor on issues such as homosexuality , single motherhood and unwed couples . Theologians cautioned that Pope Francis had spoken casually , not made a doctrinal statement . Reverend James Martin , a Jesuit priest and editor-at-large of America , a Catholic magazine , said he believed that Pope Francis was at least asserting that “ God loves and Christ redeems all of creation ” , even though conservative theologians have said paradise is not for animals . “ He said paradise is open to all creatures , ” Rev Martin said . The question of whether animals go to heaven has been debated for much of the church ’ s history . Pope Benedict said during a 2008 sermon that when an animal dies , it “ just means the end of existence on earth ” . Ms Christine Gutleben , senior director of the Humane Society , said Pope Francis ’ apparent reversal of his

predecessor ' s view could be enormous . “ If the Pope did mean that all animals go to heaven , then the implication is that animals have a soul . And if that is true , then we ought to seriously consider how we treat them because they mean something to God , ” she said . Ms Sarah Withrow King , director of Christian outreach and engagement at PETA , one of the most activist anti-slaughterhouse groups , said the Pope ' s remarks could move Catholics away from consuming meat . But there are differing views . Mr Dave Warner , a spokesman for the National Pork Producers Council , said in an email that it “ certainly does not mean that slaughtering and eating animals is a sin ” . THE NEW YORK TIMES.

**Label:** Disagree

#### Example 2

**Claim:** Back-from-the-dead Catholic priest claims God is a female.

**Evidence:** A 71 years old cleric Father John Micheal O ' neal who was officially dead for more than 48 minutes , was re-started by medics . He claims he went to heaven and met God , which he describes as a warm and comforting motherly figure . Father John Micheal O ' neal was rushed to the hospital on January 29 after a major heart attack , but was declared clinically dead soon after his arrival . With the aid of a high-tech machine called LUCAS 2 , that kept the blood flowing to his brain , doctors at Massachusetts General Hospital managed to unblock vital arteries and return his heart to a normal rhythm . However , doctors were afraid he would have suffered some brain damage from the incident , but he woke up less than 48 minutes later and seems to have perfectly recovered . The Father also claims that he has clear and vivid memories of what happened to him while he was dead .

**Label:** Agree

### Example 3

**Claim:** Soon Marijuana May Lead to Ticket , Not Arrest , in New York.

**Evidence:** After campaigning on a promise to reform stop-and-frisk , Mayor Bill de Blasio is set to launch his most significant effort to address the issues raised by the policy . Law-enforcement officials tell the New York Times that soon the NYPD may issue tickets for low-level marijuana possession rather than making arrests . Under the de Blasio administration 's planned changes to the city 's marijuana policy , those caught with a small amount of weed would be issued a court summons , but avoid a trip to the police station . The shift could have a huge impact in black and Latino communities , as a recent study found those groups represented 86 percent of those arrested for marijuana possession in the city this year . The de Blasio administration is still working out the details , such as how much weed one could possess without triggering an arrest , and some are already unhappy with the proposed change . Brooklyn District Attorney Kenneth Thompson , who announced this year that he would stop prosecuting small-scale marijuana possession cases , said that telling people to show up in a court room without arresting them may actually be worse for minorities . “ In order to give the public confidence in the fairness of the criminal justice system , these cases should be subject to prosecutorial review , ” Thompson explained . “ By allowing these cases to avoid early review , by issuing a summons , there is a serious concern that many summonses will be issued without the safeguards currently in place . These cases will move forward even when due process violations might have occurred . ” In other words , do n't expect the debate about how NYPD officers should proceed when they catch someone with a joint to be resolved anytime soon..

**Label:** Not Enough Info

## REFERENCES

- Abney, S. (2007). *Semisupervised learning for computational linguistics*. CRC Press.
- Aharoni, R. and Y. Goldberg (2020). Unsupervised domain clusters in pretrained language models. *arXiv preprint arXiv:2004.02105*.
- Alam, F., S. Joty, and M. Imran (2018). Domain adaptation with adversarial training and graph embeddings. *arXiv preprint arXiv:1805.05151*.
- Ando, R. K., T. Zhang, and P. Bartlett (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, **6**(11).
- Arjovsky, M., S. Chintala, and L. Bottou (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR.
- Axelrod, A., X. He, and J. Gao (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pp. 355–362.
- Babakar, M. and W. Moy (2016). The state of automated factchecking. *Full Fact*.
- Bahdanau, D., K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Baird, S., D. Sibley, and Y. Pan (2017). Talos targets disinformation with fake news challenge victory. *Fake News Challenge*.
- Baldwin, T., P. Cook, M. Lui, A. MacKinlay, and L. Wang (2013). How noisy social media text, how different social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 356–364.

- Bast, H., B. Buchhold, and E. Haussmann (2017). Overview of the triple scoring task at the WSDM Cup 2017. *arXiv preprint arXiv:1712.08081*.
- Beltagy, I., K. Lo, and A. Cohan (2019). Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.
- Ben-David, E., C. Rabinovitz, and R. Reichart (2020). Perl: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Transactions of the Association for Computational Linguistics*, **8**, pp. 504–521.
- Ben-David, S., J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan (2010). A theory of learning from different domains. *Machine learning*, **79**(1), pp. 151–175.
- Bengio, Y., Y. Lecun, and G. Hinton (2021). Deep learning for AI. *Communications of the ACM*, **64**(7), pp. 58–65.
- Bentivogli, L., P. Clark, I. Dagan, and D. Giampiccolo (2009). The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.
- Bentivogli, L., P. Clark, I. Dagan, and D. Giampiccolo (2011). The Seventh PASCAL Recognizing Textual Entailment Challenge. In *TAC*. Citeseer.
- Birnbaum, L. (1991). Rigor mortis: a response to Nilsson’s “Logic and artificial intelligence”. *Artificial Intelligence*, **47**(1-3), pp. 57–77.
- Blitzer, J., R. McDonald, and F. Pereira (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 120–128.
- Blum, A. and T. Mitchell (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100.

- Bohnet, B., R. McDonald, G. Simoes, D. Andor, E. Pitler, and J. Maynez (2018). Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings. *arXiv preprint arXiv:1805.08237*.
- Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250.
- Bos, J. and K. Markert (2005). Recognising textual entailment with logical inference. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 628–635. Association for Computational Linguistics.
- Bousmalis, K., G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan (2016). Domain separation networks. *Advances in neural information processing systems*, **29**, pp. 343–351.
- Bowman, S. R., G. Angeli, C. Potts, and C. D. Manning (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
- Chang, A. X. and C. D. Manning (????). Sutime: A library for recognizing and normalizing time expressions.
- Charniak, E. (1997). Statistical parsing with a context-free grammar and word statistics. *AAAI/IAAI*, **2005**(598-603), p. 18.
- Charniak, E., Y. Altun, R. de Salvo Braz, B. Garrett, M. Kosmala, T. Moscovich, L. Pang, C. Pyo, Y. Sun, W. Wy, et al. (2000). Reading comprehension programs in a statistical-language-processing class. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.
- Chen, M., Z. Xu, K. Weinberger, and F. Sha (2012). Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*.

- Chen, Q., X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen (2016). Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.
- Chen, Z., Y. Cui, W. Ma, S. Wang, T. Liu, and G. Hu (2018). HFL-RC system at SemEval-2018 task 11: hybrid multi-aspects model for commonsense reading comprehension. *arXiv preprint arXiv:1803.05655*.
- Cheng, J., L. Dong, and M. Lapata (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.
- Chklovski, T. and P. Pantel (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 33–40.
- Cho, K., B. Van Merriënboer, D. Bahdanau, and Y. Bengio (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.
- Ciampaglia, G. L., P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini (2015). Computational fact checking from knowledge networks. *PloS one*, **10**(6), p. e0128193.
- Ciaramita, M. and M. Johnson (2003). Supersense tagging of unknown nouns in WordNet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 168–175. Association for Computational Linguistics.
- Clark, K., U. Khandelwal, O. Levy, and C. D. Manning (2019). What does bert look at? an analysis of bert’s attention. *arXiv preprint arXiv:1906.04341*.
- Clinchant, S., G. Csurka, and B. Chidlovskii (2016). A domain adaptation regularization for denoising autoencoders. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 26–31.

- Condoravdi, C., D. Crouch, V. De Paiva, R. Stolle, and D. G. Bobrow (2003). Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pp. 38–45.
- Cui, X. and D. Bollegala (2019). Self-adaptation for unsupervised domain adaptation. *Proceedings-Natural Language Processing in a Deep Learning World*.
- Dagan, I., B. Dolan, B. Magnini, and D. Roth (2010). Recognizing textual entailment: Rational, evaluation and approaches—erratum. *Natural Language Engineering*, **16**(1), pp. 105–105.
- Dagan, I., O. Glickman, and B. Magnini (2005). The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pp. 177–190. Springer.
- Dagan, I., D. Roth, M. Sammons, and F. M. Zanzotto (2013). Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, **6**(4), pp. 1–220.
- Daumé III, H. (2009). Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.
- Daume III, H. and D. Marcu (2006). Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, **26**, pp. 101–126.
- Davis, E. and G. Marcus (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, **58**(9), pp. 92–103.
- Derczynski, L. and K. Bontcheva (2014). PHEME: Veracity in Digital Social Networks. In *UMAP workshops*.
- Desai, S., B. Sinno, A. Rosenfeld, and J. J. Li (2019). Adaptive ensembling: Unsupervised domain adaptation for political document analysis. *arXiv preprint arXiv:1910.12698*.

- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota. doi:10.18653/v1/N19-1423.
- Doddington, G. R., A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pp. 837–840. Lisbon.
- Dodge, J., G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Dzikovska, M. O., R. D. Nielsen, C. Brew, C. Leacock, D. Giampiccolo, L. Bentivogli, P. Clark, I. Dagan, and H. T. Dang (2013). Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, NORTH TEXAS STATE UNIV DENTON.
- Ferreira, W. and A. Vlachos (2016). Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1163–1168.
- Finkel, J. R. and C. D. Manning (2009). Nested named entity recognition. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pp. 141–150.
- for Computing Machinery, A. (1983). *Computing Reviews*, **24**(11), pp. 503–512.

- Francis, D. (2016). Fast & furious fact check challenge.
- Fyodorov, Y., Y. Winter, and N. Francez (2000). A natural logic inference system. In *Proceedings of the 2nd Workshop on Inference in Computational Semantics (ICoS-2)*. Citeseer.
- Gabrilovich, E., M. Ringgaard, and A. Subramanya (2013). FACC1: Freebase annotation of ClueWeb corpora.
- Gal, Y. and Z. Ghahramani (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR.
- Ganin, Y. and V. Lempitsky (2015). Unsupervised domain adaptation by back-propagation. In *International conference on machine learning*, pp. 1180–1189. PMLR.
- Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, **17**(1), pp. 2096–2030.
- Giampiccolo, D., H. T. Dang, B. Magnini, I. Dagan, E. Cabrio, and B. Dolan (2008). The Fourth PASCAL Recognizing Textual Entailment Challenge. In *TAC*. Citeseer.
- Giampiccolo, D., B. Magnini, I. Dagan, and W. 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.
- Glickman, O. (2006). *Applied textual entailment*. Citeseer.
- Glorot, X., A. Bordes, and Y. Bengio (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.

- Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. *Advances in neural information processing systems*, **27**.
- Gordon, A. (2016). Commonsense interpretation of triangle behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Gorrell, G., E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, and L. Derczynski (2019). SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 845–854.
- Gottipati, S., M. Qiu, L. Yang, F. Zhu, and J. Jiang (2013). Predicting user’s political party using ideological stances. In *International Conference on Social Informatics*, pp. 177–191. Springer.
- Gretton, A., K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola (2006). A kernel method for the two-sample-problem. *Advances in neural information processing systems*, **19**, pp. 513–520.
- Gretton, A., A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, **3**(4), p. 5.
- Grishman, R. and B. Sundheim (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Guo, H., R. Pasunuru, and M. Bansal (2020). Multi-source domain adaptation for text classification via distancenet-bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7830–7838.
- Gururangan, S., A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith (2020). Don’t stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

- Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith (2018a). Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112. Association for Computational Linguistics, New Orleans, Louisiana. doi:10.18653/v1/N18-2017.
- Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith (2018b). Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.
- Habash, N. and B. Dorr (2003). CATVAR: A database of categorial variations for English. In *Proceedings of the MT Summit*, pp. 471–474. Citeseer.
- Haim, R. B., I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor (2006). The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.
- Han, X. and J. Eisenstein (2019). Unsupervised domain adaptation of contextualized embeddings for sequence labeling. *arXiv preprint arXiv:1904.02817*.
- Hanselowski, A., P. Avinesh, B. Schiller, and F. Caspelherr (2017). Description of the system developed by team athene in the fnc-1. *Fake News Challenge*.
- Hassan, N., F. Arslan, C. Li, and M. Tremayne (2017). Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1803–1812.
- Hassan, N., C. Li, and M. Tremayne (2015). Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th acm international on conference on information and knowledge management*, pp. 1835–1838.

- Henaff, M., J. Weston, A. Szlam, A. Bordes, and Y. LeCun (2016). Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*.
- Hendrycks, D., X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song (2020). Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.
- Hickl, A., J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi (2006). Recognizing textual entailment with LCC’s GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop*, volume 18.
- Hinton, G., O. Vinyals, and J. Dean (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Hirschman, L., M. Light, E. Breck, and J. D. Burger (1999). Deep read: A reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pp. 325–332.
- Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation*, **9**(8), pp. 1735–1780.
- Howard, J. and S. Ruder (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- Hua, X. and L. Wang (2017). Understanding and detecting supporting arguments of diverse types. *arXiv preprint arXiv:1705.00045*.
- Jia, C., X. Liang, and Y. Zhang (2019). Cross-domain NER using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2464–2474.
- Jiang, J. (2008). A literature survey on domain adaptation of statistical classifiers. URL: <http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey>, **3**(1-12), p. 3.

- Jiang, J. and C. Zhai (2007). Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 264–271.
- Kamath, A. and R. Das (2018). A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.
- Khot, T., A. Sabharwal, and P. Clark (2018). Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Kim, S., J.-H. Hong, I. Kang, and N. Kwak (2018). Semantic sentence matching with densely-connected recurrent and co-attentive information. *arXiv preprint arXiv:1805.11360*.
- Kim, S., I. Kang, and N. Kwak (2019). Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 6586–6593.
- Kim, Y.-B., K. Stratos, and D. Kim (2017). Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1297–1307.
- Kovaleva, O., A. Romanov, A. Rogers, and A. Rumshisky (2019). Revealing the dark secrets of BERT. *arXiv preprint arXiv:1908.08593*.
- Krueger, D., E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville (2021). Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR.
- Lai, A. and J. Hockenmaier (2014). Illinois-LH: A Denotational and Distributional Approach to Semantics. In *SemEval@ COLING*, pp. 329–334.
- Laine, S. and T. Aila (2016). Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

- Laine, S. M. and T. O. Aila (2018). Temporal ensembling for semi-supervised learning. US Patent App. 15/721,433.
- Lakoff, G. (1970). Linguistics and natural logic. *Synthese*, **22**(1), pp. 151–271.
- Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), pp. 1234–1240.
- Li, Z., Y. Wei, Y. Zhang, and Q. Yang (2018). Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Li, Z., Y. Zhang, Y. Wei, Y. Wu, and Q. Yang (2017). End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification. In *IJCAI*, pp. 2237–2243.
- Lim, K., J. Y. Lee, J. Carbonell, and T. Poibeau (2020). Semi-supervised learning on meta structure: Multi-task tagging and parsing in low-resource scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8344–8351.
- Ling, X. and D. S. Weld (2012). Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- Long, Y. (2017). Fake news detection through multi-perspective speaker profiles. Association for Computational Linguistics.
- MacCartney, B. and C. D. Manning (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 193–200.
- MacCartney, B. and C. D. Manning (2009). An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*, pp. 140–156. Association for Computational Linguistics.

- Manning, C., M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60.
- Marcus, M., B. Santorini, and M. A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank.
- Marelli, M., S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli, et al. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Lrec*, pp. 216–223. Reykjavik.
- McCarthy, J. et al. (1960). *Programs with common sense*. RLE and MIT computation center.
- McClosky, D., E. Charniak, and M. Johnson (2006). Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 337–344.
- Mikolov, T., E. Grave, P. Bojanowski, C. Puhersch, and A. Joulin (2017). Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, **38**(11), pp. 39–41.
- Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, **3**(4), pp. 235–244.
- Miller, T. (2019). Simplified neural unsupervised domain adaptation. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2019, p. 414. NIH Public Access.

- Minsky, M. (2000). Commonsense-based interfaces. *Communications of the ACM*, **43**(8), pp. 66–73.
- Mintz, M., S. Bills, R. Snow, and D. Jurafsky (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1003–1011.
- Moore, R. C. (1982). *The role of logic in knowledge representation and commonsense reasoning*. SRI International. Artificial Intelligence Center.
- Moore, R. C. and W. Lewis (2010). Intelligent selection of language model training data.
- Morgenstern, L. and C. Ortiz (2015). The winograd schema challenge: Evaluating progress in commonsense reasoning. In *Twenty-Seventh IAAI Conference*.
- Mostafazadeh, N., N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen (2016). A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849.
- Nakashole, N. and T. Mitchell (2014). Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1019.
- Ng, H. T., L. H. Teo, and J. L. P. Kwan (2000). A machine learning approach to answering questions for reading comprehension tests. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 124–132.
- Nie, Y., H. Chen, and M. Bansal (2018). Combining Fact Extraction and Verification with Neural Semantic Matching Networks. *arXiv preprint arXiv:1811.07039*.

- Nunberg, G. (1987). Position paper on common-sense and formal semantics. In *Theoretical Issues in Natural Language Processing 3*.
- Oren, Y., S. Sagawa, T. B. Hashimoto, and P. Liang (2019). Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060*.
- Pan, S. J., X. Ni, J.-T. Sun, Q. Yang, and Z. Chen (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pp. 751–760.
- Pan, S. J. and Q. Yang (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, **22**(10), pp. 1345–1359.
- Panenghat, M. P., S. Suntwal, F. Rafique, R. Sharp, and M. Surdeanu (2020). Towards the Necessity for Debiasing Natural Language Inference Datasets. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 6883–6888.
- Parikh, A. P., O. Täckström, D. Das, and J. Uszkoreit (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.
- Peirce, C. S. (1883). A theory of probable inference.
- Peng, N. and M. Dredze (2016). Multi-task domain adaptation for sequence tagging. *arXiv preprint arXiv:1608.02689*.
- Pennington, J., R. Socher, and C. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.
- Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Petrov, S., P.-C. Chang, M. Ringgaard, and H. Alshawi (2010). Uptraining for accurate deterministic question parsing.

- Petrov, S. and R. McDonald (2012). Overview of the 2012 shared task on parsing the web.
- Peyrard, M. (2019). A simple theoretical model of importance for summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1059–1073.
- Phang, J., I. Calixto, P. M. Htut, Y. Pruksachatkun, H. Liu, C. Vania, K. Kann, and S. R. Bowman (2020). English intermediate-task training improves zero-shot cross-lingual transfer too. *arXiv preprint arXiv:2005.13013*.
- Phang, J., T. Févry, and S. R. Bowman (2018). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.
- Plank, B. (2011). *Domain adaptation for parsing*. Citeseer.
- Plank, B., A. Johannsen, and A. Søgaard (2014). Importance weighting and unsupervised domain adaptation of POS taggers: a negative result. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 968–973.
- Plank, B. and G. Van Noord (2011). Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1566–1576.
- Poliak, A., J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme (2018). Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.
- Pomerleau, D. and D. Rao (2017). Fake news challenge.
- Radford, A., K. Narasimhan, T. Salimans, and I. Sutskever (2018). Improving language understanding by generative pre-training.

- Raina, R., A. Y. Ng, and C. D. Manning (2005). Robust textual inference via learning and abductive reasoning. In *AAAI*, pp. 1099–1105.
- Rashkin, H., E. Choi, J. Y. Jang, S. Volkova, and Y. Choi (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 2931–2937.
- Rashkin, H., M. Sap, E. Allaway, N. A. Smith, and Y. Choi (2018). Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939*.
- Rasmus, A., H. Valpola, M. Honkala, M. Berglund, and T. Raiko (2015). Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*.
- Riedel, B., I. Augenstein, G. P. Spithourakis, and S. Riedel (2017). A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. *arXiv preprint arXiv:1707.03264*.
- Romanov, A. and C. Shivade (2018). Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*.
- Rotman, G. and R. Reichart (2019). Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, **7**, pp. 695–713.
- Ruder, S. (2019). *Neural transfer learning for natural language processing*. Ph.D. thesis, NUI Galway.
- Ruder, S. and B. Plank (2017). Learning to select data for transfer learning with bayesian optimization. *arXiv preprint arXiv:1707.05246*.
- Ruder, S. and B. Plank (2018). Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*.

- Saito, K., Y. Ushiku, and T. Harada (2017). Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 2988–2997. PMLR.
- Sajjadi, M., M. Javanmardi, and T. Tasdizen (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, **29**, pp. 1163–1171.
- Schneider, N. and N. A. Smith (2015). A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1537–1547.
- Schuster, T., D. J. Shah, Y. J. S. Yeo, D. Filizzola, E. Santus, and R. Barzilay (2019). Towards debiasing fact verification models. *arXiv preprint arXiv:1908.05267*.
- Shah, D. J., T. Lei, A. Moschitti, S. Romeo, and P. Nakov (2018). Adversarial domain adaptation for duplicate question detection. *arXiv preprint arXiv:1809.02255*.
- Shen, J., Y. Qu, W. Zhang, and Y. Yu (2018). Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Shi, G., C. Feng, L. Huang, B. Zhang, H. Ji, L. Liao, and H.-Y. Huang (2018). Genre separation network with adversarial training for cross-genre relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1018–1023.
- Silverman, C. (2015). Lies, damn lies and viral content.
- Søgaard, A. and C. Rishøj (2010). Semi-supervised dependency parsing using generalized tri-training. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1065–1073.

- Steedman, M., R. Hwa, S. Clark, M. Osborne, A. Sarkar, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim (2003). Example selection for bootstrapping statistical parsers. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 236–243.
- Sun, K., D. Yu, D. Yu, and C. Cardie (2018). Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*.
- Sundheim, B. M. (1993). TIPSTER/MUC-5 information extraction system evaluation. Technical report, NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT AND E DIV SAN DIEGO CA.
- Suntwal, S., M. Paul, R. Sharp, and M. Surdeanu (2019a). On the Importance of Delexicalization for Fact Verification. 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)*, pp. 3413–3418. Association for Computational Linguistics, Hong Kong, China. doi:10.18653/v1/D19-1340.
- Suntwal, S., M. Paul, R. Sharp, and M. Surdeanu (2019b). On the Importance of Delexicalization for Fact Verification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, (Short Papers)*. Association for Computational Linguistics, Hong Kong.
- Tarvainen, A. and H. Valpola (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pp. 1195–1204.
- Thorne, J. and A. Vlachos (2017). An extensible framework for verification of numerical claims. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 37–40.

- Thorne, J., A. Vlachos, C. Christodoulopoulos, and A. Mittal (2018a). FEVER: a large-scale dataset for Fact Extraction and VERification. *arXiv preprint arXiv:1803.05355*.
- Thorne, J., A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal (2018b). The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp. 1–9. Association for Computational Linguistics, Brussels, Belgium. doi: 10.18653/v1/W18-5501.
- Tsuchida, M. and K. Ishikawa (2011). IKOMA at TAC2011: A Method for Recognizing Textual Entailment using Lexical-level and Sentence Structure-level features. In *TAC*.
- Tsvetkov, Y., M. Faruqui, W. Ling, B. MacWhinney, and C. Dyer (2016). Learning the curriculum with bayesian optimization for task-specific word representation learning. *arXiv preprint arXiv:1605.03852*.
- Turc, I., M.-W. Chang, K. Lee, and K. Toutanova (2019). Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.
- Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103.
- Vlachos, A. and S. Riedel (2014). Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pp. 18–22.

- Vlachos, A. and S. Riedel (2015). Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2596–2601. Association for Computational Linguistics.
- Volpi, R., H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese (2018). Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*.
- Vosoughi, S., D. Roy, and S. Aral (2018). The spread of true and false news online. *Science*, **359**(6380), pp. 1146–1151.
- Wang, A., A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Wang, W. Y. (2017). ” liar, liar pants on fire”: A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.
- Weston, J., A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.
- Weston, J., S. Chopra, and A. Bordes (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.
- Williams, A., N. Nangia, and S. R. Bowman (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. (2019). HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *ArXiv*, pp. arXiv–1910.

- Yadav, V., S. Bethard, and M. Surdeanu (2019). Alignment over Heterogeneous Embeddings for Question Answering. 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)*, pp. 2681–2691.
- Yang, Y. and J. Eisenstein (2014). Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 538–544.
- Zadeh, L. A. (1988). Fuzzy logic. *Computer*, **21**(4), pp. 83–93.
- Zeman, D. and P. Resnik (2008). Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.
- Zhang, C. and J. Chai (2010). Towards conversation entailment: An empirical investigation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 756–766.
- Zhang, S., R. Rudinger, K. Duh, and B. Van Durme (2017). Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, **5**, pp. 379–395.
- Zhou, Z.-H. and M. Li (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, **17**(11), pp. 1529–1541.
- Zhu, X. and A. B. Goldberg (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, **3**(1), pp. 1–130.
- Ziser, Y. and R. Reichart (2016). Neural structural correspondence learning for domain adaptation. *arXiv preprint arXiv:1610.01588*.

- Ziser, Y. and R. Reichart (2018a). Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 238–249.
- Ziser, Y. and R. Reichart (2018b). Pivot based language modeling for improved neural domain adaptation. 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. 1241–1251.
- Ziser, Y. and R. Reichart (2019). Task refinement learning for improved accuracy and stability of unsupervised domain adaptation. In *proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pp. 5895–5906.