Causal Relation Extraction: a Three Step Process

A Thesis
Presented to
The Division of Mathematical and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Kai Pinckard

May 2021

Approved for the Division
(Computer Science)

____

Eitan Frachtenberg

Acknowledgements

To Eitan, thank you for your support and guidance throughout the process of writing this thesis. I greatly enjoyed working with you, and I learned a lot from you.

To Mark, thank you for always letting me in your classes from the wait-list, and thank you for including me in your research over the summer. It is from these experiences that I learned much of what I know about deep learning.

To my housemates at The Zone, thank you for being so fun and supportive. Between the smoke from the fires, COVID, and losing power in a winter storm, to name a few things, it has been a difficult year but you guys made it much easier.

To the Gillespie Family Student Research Fund, thank you for so generously funding my Reed College Initiative grant. Without the GPU this grant enabled Reed to purchase, it would not have been possible to train the neural networks I used in this thesis investigation.

To my family, thank you for your loving support and for always believing in me.

List of Abbreviations

AI	Artificial Intelligence
ALBERT	A Lite BERT
BERT	Bidirectional Encoder Representations from Transformers
GPT-2	Generative Pre-trained Transformer 2
NLP	Natural Language Processing
RoBERTa	Robustly Optimized BERT Approach
VRAM	Video Random Access Memory

Table of Contents

List of Tables

List of Figures

Abstract

Extracting cause-effect relationships from natural language texts remains a challenging problem in the field of artificial intelligence. In this thesis investigation, the task of extracting cause-effect relationships is broken down into three binary classification problems. Each of these steps is performed by a fine-tuned neural network. In the first step, a neural network determines whether or not a given sentence contains a cause-effect relationship. In the second step, a neural network determines which pairs of nouns in the sentence are causally related. Finally, in the third step, a neural network identifies the direction of the cause-effect relationship between a given pair of nouns. The classifier neural networks achieved a promising level of accuracy on each of the three steps.

Introduction

0.1 What is Causal Relation Extraction and Why Does it Matter?

0.1.1 What is a Causal Relation?

A causal relation is a cause-effect relationship. Generally, a cause-effect relationship is thought of as relating two events X and Y such that X caused Y to happen. However, there is a wide variety of definitions for a causal relationship. The SemEval-2010 Task 8 data set used in this investigation defines a causal relation as occurring whenever "An event or object yields an effect" (Hendrickx et al. (2010b)). Others define a causal relation in terms of counterfactuals, such that the statement "X caused Y" implies that if X had not occurred then Y would also not have occurred (Starr (2021)). Still others define causal relations to include situations where the occurrence of X greatly increases the probability of the occurrence of Y. Thus, there is no broad consensus on the specific definition of a causal relationship. As a result, there is often ambiguity in terms of what exactly is expressed by a sentence containing a causal relation.

Furthermore, non-technical writers frequently express different subtypes of causal relations inconsistently or neglect to identify what type of causal relation is being expressed. According to the Harvard Graduate School of Education "We tend to use simple patterns–where a cause directly leads to an effect–to explain what happens in our world (Group (2010)). But the world is seldom so simple!" The Harvard Graduate School of education presents six subtypes of causal relations commonly occurring in science: linear causality, domino causality, cyclic causality, spiraling causality, relational causality, and mutual causality (Group (2010)). A causal relation is linear when there is a direct link between one cause and one effect. In contrast, domino causality occurs when the effect in a linear casual pattern is the cause in a second linear casual pattern. Frequently, sentences expressing domino causality omit the intermediate effect(s) (Group (2010)). For example, in the sentence "The stove caused the water in the pot to heat up". This sentence is expressing a linear causal relationship between the stove and the water in the pot. However, in reality the stove causes the pot to heat up and the pot causes the water to heat up. Understanding that the sentence is expressing a domino type causal relation requires world knowledge. Similar ambiguities can arise when expressing causal relationships of each of the other four causal patterns. Thus, understanding exactly what is meant by a particular causal relation in a body of text is nontrivial and often requires interpretation.

0.1.2 How are Causal Relations Expressed in Natural Language Texts?

In the paper *Expressing Causation in Written English* (1992), a total of 130 devices for expressing causation were collected. The investigation's motivation was to determine which causative devices should be included in school curricula to teach students to better identify and understand causal relationships (Fang Xuelan & Kennedy (1992)). Furthermore, the paper analysed the frequency with which these devices were used in the one million word Lancaster-Oslo-Bergen (LOB) Corpus so that devices with higher usage frequencies could be prioritized in lesson plans. For a complete list of the 130 causative devices examined see the causative devices table in Appendix A.

Overall, the study identified 11 major categories of causative devices. Of these categories only the 8 shown in Figure 1 below explicitly signal the presence of a causal relation with causative markers such as "cause", "from" or "because of". For example, in the sentence "Britain suffers comparatively little from the effect of the pollution in the North sea because of winds, currents, depths and the nature of its rivers." (Fang Xuelan & Kennedy (1992)).

The other 3 major categories only implicitly signalled the presence of a causal relation. The three non-explicit categories of causal devices identified are implicit causative verbs, elliptical syntactic patterns, and juxtaposition (Fang Xuelan & Kennedy (1992)). An example of a sentence with implicit causative verbs is "He made me sad." The word "made" is considered to be an implicit causative verb because it could not be replaced with an explicit causative verb phrase such as "cause", "produce" or "result from" (Fang Xuelan & Kennedy (1992)). For an example of a sentence with an elliptical syntactic pattern consider the following sentence: "Being Christmas, the library was closed" (Fang Xuelan & Kennedy (1992)). This sentence implies that it being Christmas caused the library to be closed. For an example of the juxtaposition as an implicit causative device consider the following sentences: "It is raining hard.

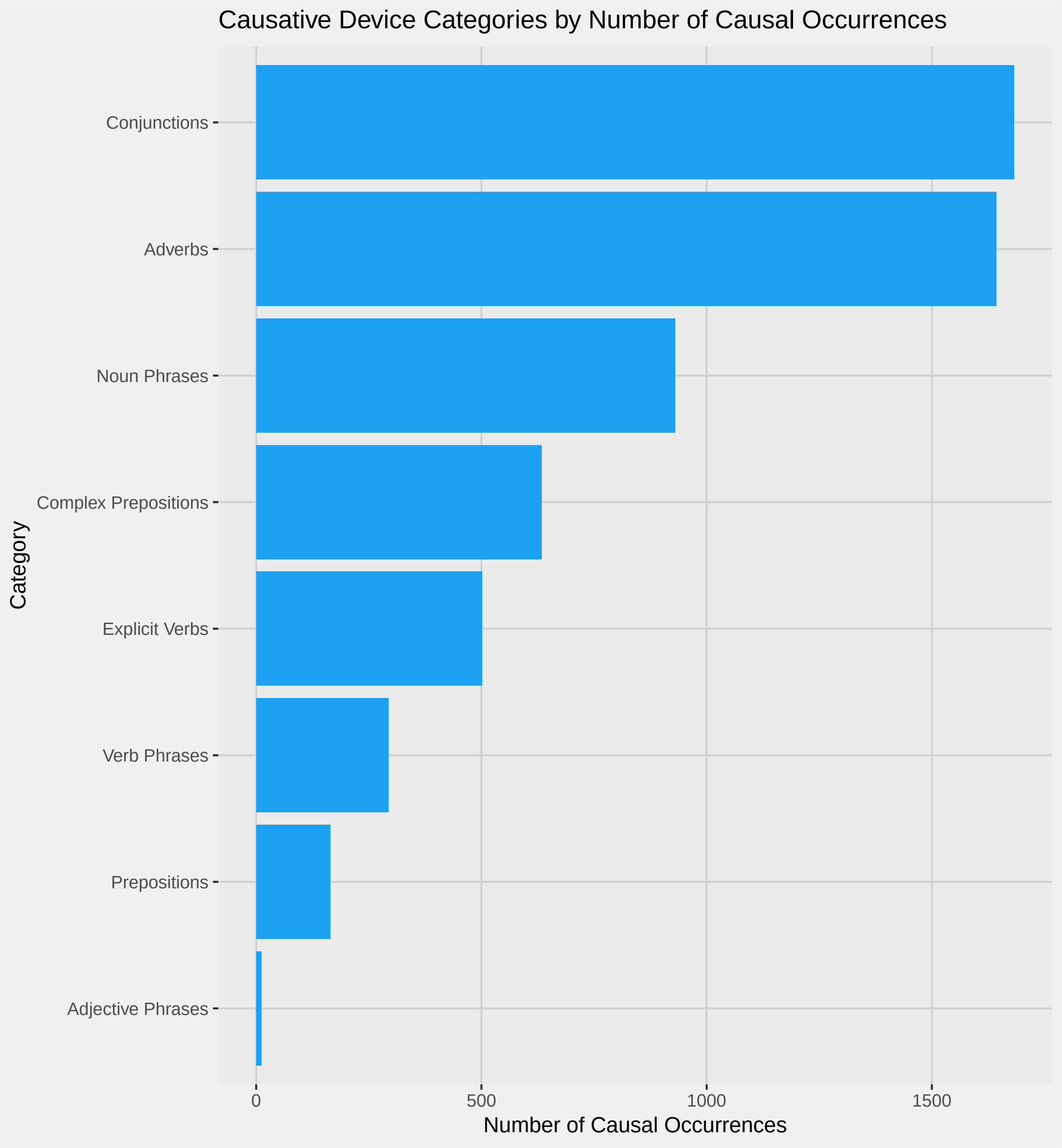

Figure 1: Causative Device Categories by Number of Occurrences

We should cancel the picnic" (Fang Xuelan & Kennedy (1992)). It is implied that it being the case that it was raining hard caused the sentence's writer to want to cancel the picnic. These three categories of causative devices were briefly discussed here to provide a more complete description of the diverse ways in which causal relations are expressed in the English language. However, because they often ambiguously express causal relations they are largely disregarded for the remainder of this investigation.

0.1.3 What is Causal Relation Extraction?

Causal relation extraction is the process of identifying and extracting causal relations from natural language texts. An extracted causal relation consists of a cause and an effect. The extracted causal relations can be stored for use in other tasks. For example, consider the sentence "Eating candy caused the boy to feel happy". The extracted causal relation might be stored as "eating candy ->the boy to feel happy".

0.1.4 Why is it Hard to Extract Causal Relations?

The first reason it is hard to extract causal relations is that there is a wide variety of ways that causal relations are defined and used by different individuals. Secondly, despite there being a large number of different definitions for causal relations, there is arguably an even larger number of ways that each individual definition is expressed in English. Finally, a significant body of research shows that young children do not consistently identify causal relations in narratives until they have acquired sufficient knowledge of human intentions, plans, actions, and outcomes (Brown (2008)). This implies that world knowledge is sometimes necessary to correctly identify causal relations in text. For these three reasons, the task of extracting causal relations from text written by individuals to be read by other individuals remains a challenging and open problem in the field of artificial intelligence.

0.1.5 Why is Extracting Causal Relations Useful?

Understanding Causal Relations is a Key Aspect of Human Reasoning

From an early age, children learn that causal relations are of critical importance (Brown (2008)). Furthermore, research has shown that 8-year-old and 10-year-old children's understanding of causal relations was a strong predictor of their later reading comprehension skills (Kendeou et al. (2009)). For this reason, investigations like the paper *Expressing Causation in Written English* (1992) have been conducted on how to best teach children about causal relations.

Beyond the importance of causal relations to children, research on reading, writing, and storytelling demonstrates that individuals place more attention on cause-effect relationships than other information, suggesting they view it as more important (Lorch et al. (1999a); Trabasso & van den Broek (1985)). Furthermore, causally related events within a story's plot were recalled better both shortly after and long after a story was read when compared to non-causally related events (Brown (2008)). Moreover, causal relations have been found to enhance recall of information in discourse (Lorch et al. (1999a,b); Trabasso & van den Broek (1985)), and reading (Brown (2008)). Since numerous studies have shown that individuals both pay more attention to and remember causal information better than non-causal information, it seems likely that the brain is biologically predisposed to pay more attention to causal information.

Finally, a significant body of research has demonstrated that understanding causal relations is of critical importance in a wide variety of life skills (Brown (2008)). A solid understanding of causal relationships is a crucial aspect of analogical reasoning (McGill (2002); Brown (2008)), causal reasoning, and decision making (Perales et al. (2004); Brown (2008)). Furthermore, research has demonstrated that one's understanding of causal relations guides one's writing (Brown (2008)).

Why Build a Knowledge Base?

A knowledge base is a structured collection of information. Knowledge bases have been successfully used to construct expert systems in the past such as MYCIN, a rule-based system for medical diagnosis and therapy (Engelmore (1987)). However, for AI systems that use knowledge bases to be effective they require large quantities of knowledge to be added to the knowledge base (Richardson & Domingos (2003)). Extracting this knowledge and entering it into a knowledge base has long been the major bottleneck holding back these types of AI systems (Richardson & Domingos (2003)). Therefore, the process of automatically populating a knowledge base with facts extracted from natural language texts is of great value and has received tremendous interest from academia (Niu et al. (2012)). Furthermore, in recent years, the amount of information available to be parsed on the internet has exploded. A person alive today can find the answer to almost any question in an instant. If AI systems could be granted access to this information through an automated information extraction system, the benefits would be enormous (Weikum & Theobald (2010)). In this investigation, the focus is on the automatic extraction of causal relations from natural language texts. If a sufficiently powerful causal relation extraction system existed, then a large knowledge base of cause-effect relationships could be automatically created. Such a knowledge base would be helpful in a wide variety of tasks. For example, such a knowledge base could be used to improve an AI system's ability to reason about causal relations with incomplete information, as demonstrated in the following example involving answering a question.

Question:
"The man flipped the main circuit breaker in his house to the off setting. The man's microwave turned off. How can he turn it on?"

When answering this question, an AI system may respond that the man needs to press his microwave's "On" button to turn it on. However, a person would know that the microwave can not be turned on until the main circuit breaker is switched back on. The AI system will likely not know that this is necessary because it is not explicitly stated in the question. However, a knowledge base of causal relations could be used to supply the AI system with the required information to correctly answer this question. Since the knowledge base would be constructed by extracting the causal relations contained in a huge collection of reputable sources it would likely contain a relation stating that if a house's main circuit breaker is turned off all devices in the house will lose power. Furthermore, it would likely contain the relation that a device being unpowered causes it to stop working. From here a set of basic logic rules and a model performing co-reference resolution could be used to determine that the microwave will not power on until the main circuit breaker is switched back on.

Causal Relations provide a Useful Constraint for Narrative Generation

Generating consistent and compelling narratives remains on the frontier of text generation technology (Fan et al. (2018)). Although recent advances in the field of artificial intelligence have resulted in improved thematic and stylistic coherence, it is still common for text to be generated with logical inconsistencies (Mao et al. (2020)). For example, "It was a sunny, warm summer night" (Mao et al. (2020)).

Even in humans understanding, grammar alone does not seem to be enough to avoid this issue. A significant body of research shows that preschool age children do not consistently notice the cause-effect relationships between story events (Brown (2008)). However, children's ability to understand causal relations linking story events has been found to improve with age, reflecting an increase in knowledge of human intentions, plans, actions, and outcomes (Brown (2008)). The implication is that world knowledge is needed to produce causally coherent narratives (Brown (2008)).

Logical inconsistencies such as in the sentence "It was a sunny, warm summer night" (Mao et al. (2020)) could be reduced by leveraging a knowledge base in the text generation process. The knowledge base could be used to check that the situations described in generated text are consistent with a large set of basic facts about the world. For example, the fact that it is not sunny at night.

Another challenge in generating long narrative texts is that stories must remain consistent throughout their entirety, requiring the modeling of very long range dependencies (Fan et al. (2018)). When attempting to generate long narratives with AI systems such as neural networks it is common for text to shift from topic to topic, lacking the maintained focus observed in human-written texts. For example, when given the story prompt "The silence before the storm comes" the AI system known as GPT-2 generated the following text.

"The silence before the storm comes as a new study, published by the American Academy of Pediatrics, finds that a child 's own silence in an incident may actually be more dangerous than the incident itself. The silent storm may come from an act of protest. It may not. It could be as subtle as a hand-drawn sign on a tree. A letter written in blood . A child's name etched into a tree. (The song "A Song for Laura" by The Beatles.) You are so much afraid"

In this example, the text started by describing a new study before quickly going off topic and discussing an unrelated and non-existent song by the Beatles. In contrast, human-written texts tend to maintain focus on a single topic for much longer. It appears that one of the reasons human-written texts tend to be more focused is that human-writers are likely to base their continuation of a given text based on cause-effect relationships that have been introduced earlier in the text (Brown (2008)). A study conducted by Van den Broek et al. examined the role of cause-effect relationships in the composition of written narratives (Brown (2008)). It was found that about 86% of the time participants' continuation of a narrative was causally related to the previous sentence (Brown (2008)). The authors concluded that cause-effect relationships provide a useful constraint in narrative production." (Brown (2008)).

A knowledge base of cause-effect relationships could be used to provide this useful constraint on narrative production to text generating AI systems. This could aid AI systems in maintaining focus in their writing for longer, leading to more natural sounding generated text.

Causal Relations can help Limit False but Believable Outputs by Generative Neural Networks

Another potential use for a knowledge base is to help restrict AI systems from generating text containing plausible but false information. For example, In the previous section the example generated text concluded with “(The song ”A Song for Laura” by The Beatles)”. However, upon further research it was found that The Beatles have never made a song called “A Song for Laura”.

Currently there is nothing preventing models from outputting false yet believable answers to questions. For example, when the GPT-2 text-generating AI system was prompted to complete the phrase “The man’s cancer was caused by” its completion was “an infection with a virus known as H5N”. Examining this output, the first question that comes to mind is “What is H5N?”. However, upon further investigation one finds that H5N is not a real virus. The AI system came up with the fake virus in its best effort to plausibly complete the sentence.

While both of these examples of plausible but false text generation were detected, it is likely that when these systems are in public use their outputs will not always be fact checked. This could result in the AI systems misleading the public on any number of issues from politics to treatment recommendations, depending on the task for which the AI system is generating text. Thus, restricting the generated text to only express true information would be valuable. This could be achieved by restricting the AI system to only generating text that expresses knowledge contained in its knowledge base.

Chapter 1

Background

1.1 Historical Attempts at Extracting Cause-Effect Relationships

Early attempts at automatically extracting causal relations from natural language texts such as books and articles began in the 1980s (Asghar (2016)). These early attempts used human-written linguistic rules. Furthermore, the rules were tailored for particular document topics, such as legal and medical documents (Asghar (2016)). Thus, linguistic rules for one topic could usually not be used to extract causal relations occurring in another subject (Asghar (2016)). The advantages of human-written linguistic rules are that they require relatively little computing power and data. Although these attempts were able to extract some cause-effect relationships, the high cost of writing different linguistic rules for each type of document largely rendered these methods impractical.

As the amount of data and computing power increased, a subset of artificial intelligence known as machine learning gained popularity. Machine learning focuses on systems that learn from data. In the context of causal relation extraction, machine learning allowed researchers to extract cause-effect relationships without needing to craft linguistic rules. The AI system could learn how to extract cause-effect relationships from labeled data.

1.2 Neural Networks

1.2.1 Why use Neural Networks to Extract Cause-Effect Relationships?

A neural network is a machine learning technique that has achieved state of the art results on a wide variety of tasks. A neural network achieves these impressive results by training on a data set. Researchers are now able to train neural networks that can perform tasks such as accurately classifying images into hundreds of different categories, predicting the shape a protein will fold into, or summarizing an article (Pham et al. (2021); Senior et al. (2020); Cachola et al. (2020)). Since neural networks are successfully performing many challenging tasks, they appear to be a strong candidate for an AI system extracting cause-effect relations from text. Thus, this investigation applied neural networks to the task of causal relation extraction.

1.2.2 Historical Background for Neural Networks

Neural networks were first created in the 1950s in an attempt to model the brain (Goodfellow et al., 2016, 14). Early researchers noticed that the brain contains a vast network of neurons joined together by connections called synapses. These researchers wanted neural networks to replicate this structure as closely as possible. However, due to the limited capabilities of computers at the time, the first neural networks, called perceptrons, only had a single neuron. As a result, perceptrons were severely limited in their capabilities. However, later researchers learned that increasing the size of neural networks allowed them to overcome these limitations. As a result, neural networks now have large numbers of neurons organized into multiple layers.

1.2.3 How Neural Networks Work - A High Level Description

The perceptron is the simplest kind of neural network. It contains only a single neuron and is typically used for binary classification. As shown in Figure 1.1, perceptrons can receive multiple inputs. Additionally, an input of constant value known as the bias is provided. The bias gives the perceptron the ability to bias itself towards classifying one way or the other. After receiving its inputs, the perceptron computes the weighted sum of each of the inputs. Each input's corresponding weight is represented on the connection between the input and the output neuron in Figure 1.1. This weighted

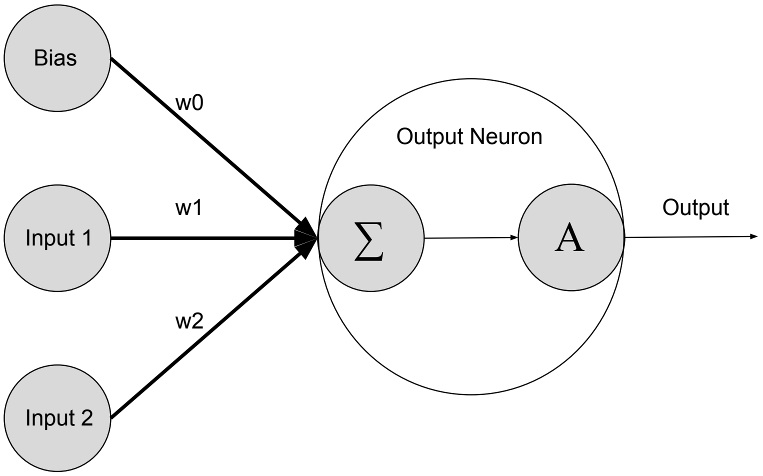

Figure 1.1: A perceptron with two inputs

sum is then fed into the activation function, represented by “A” below. Typically, in a perceptron performing binary classification, the activation function maps its input to values in the range of 0 to 1. With values close to 0 corresponding to one of the two classes and values close to 1 corresponding to the other.

A perceptron is able to correctly classify observations that are linearly separable. In this way, perceptrons are similar to logistic regression. Figure 1.2 provides an example of linearly separable observations. Specifically, it shows how a salt-loving, pepper-hating individual might review an assortment of foods.

However, like logistic regression, perceptrons are limited in that they can not solve nonlinear problems. This is a significant limitation because many important classification problems have nonlinear boundaries between the classes. For example, an AI powered cooking robot creating recipes might need to predict if recipes will be

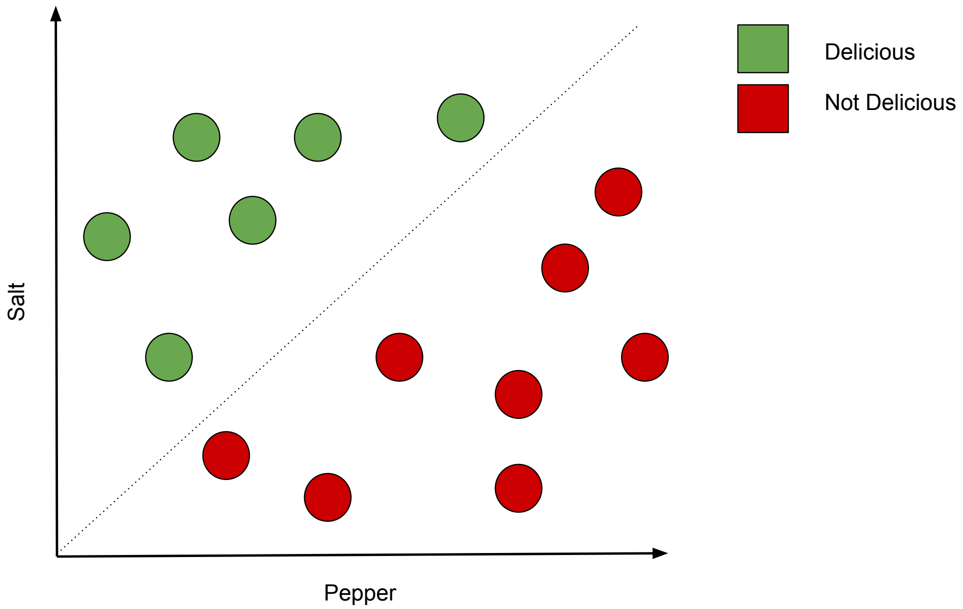

Figure 1.2: A linearly separable classification problem

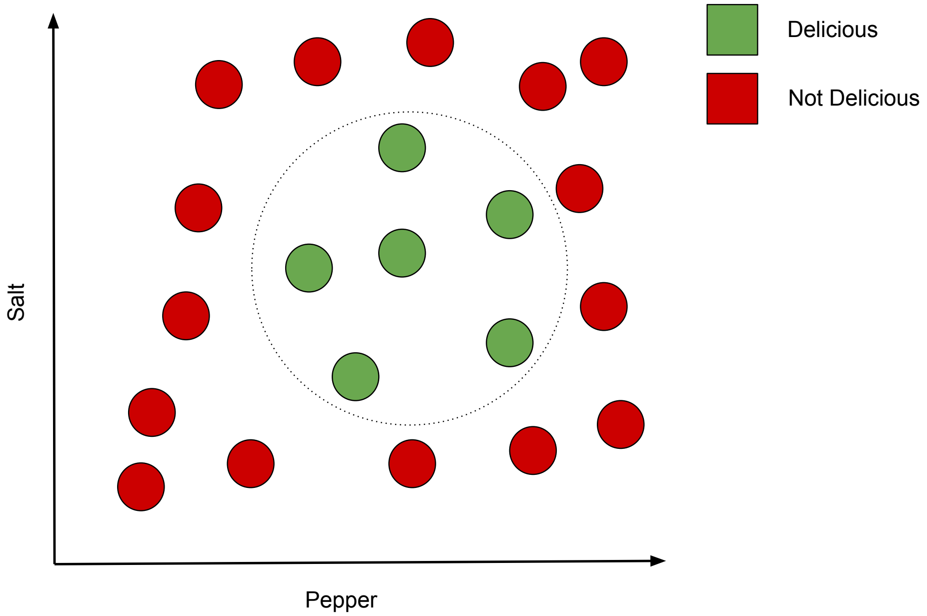

Figure 1.3: A nonlinear classification problem

delicious or not based on the amount of salt and pepper in the recipe. This relationship is nonlinear because for most individuals there is an approximately correct amount of salt and pepper for a particular dish. Slightly more or less of either is fine, but straying too far from the correct amount renders the dish “Not Delicious”. This situation is shown in Figure 1.3.

By adding additional neurons and layers to a perceptron, it gains the ability to classify observations into classes with nonlinear boundaries. Figure 1.4 shows a simple neural network with two layers and three neurons. The inputs do not count as neurons since they do not have tunable parameters. The type of neural network shown in Figure 1.4 is known as a feed forward neural network. A feed forward neural network contains an input layer, where the network first receives the input data. Each of the inputs is connected to each of the neurons in the next layer of the network. The neurons in this layer will then feed their outputs forward into each of the neurons in the next layer of the network. This process continues until the output layer of the network is reached. At this point, the final output or prediction is produced.

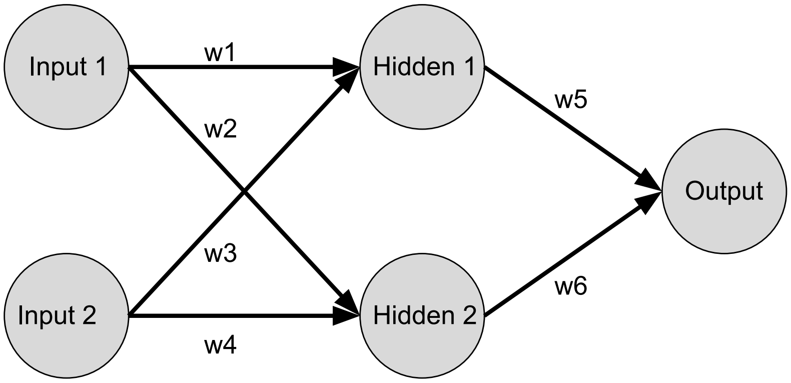

Figure 1.4: A simple feed forward neural network

1.2.4 What do Neural Networks Learn From?

Neural networks learn how to perform a task by training on a collection of data called the training data or training set. The training data will typically contain a large amount of labeled data. For example, a data set containing images with corresponding labels indicating if the image is of a cat or a dog could be used to train a neural network to predict if a previously unseen image is of a cat or a dog.

1.2.5 Important Factors when Creating or Selecting a Data Set

When creating a data set there are several important factors to keep in mind. First, the larger the data set the better. Increasing the size of the training data set has been found to increase the accuracy of neural networks in a wide variety of tasks. Even neural networks trained on data sets containing billions of words see improved accuracy when given more training data (Liu et al. (2019)).

Secondly, the data should be high quality. In the example of the cats and dogs data set this would mean having all of the image labels correctly describe what is actually contained in the image. This is important because the neural network will “trust” the labels completely. Thus, an image of a cat with the dog label, will be considered to be a dog in the network’s “eyes” and the neural network will adjust its “understanding” of a dog, so that it will be able to classify other images of cats like this ones as being dogs as well.

Thirdly, the data set should be unbiased unless a biased network is desired. An example of a highly biased data set would be if the cats and dogs data set contained 99 images of cats and 1 image of a dog. In this case, the neural network could achieve 99% accuracy by simply predicting that every input image it receives is an image of a cat. The data set should be as representative as possible of the data that the network will see when it is being used in production.

1.2.6 How do Neural Networks Learn from Data Sets?

Once a data set has been acquired, it can be used to train a neural network. When training, the network will be given each of the elements of the data set and asked to make a prediction based on each of the elements. By comparing the element’s label with the network’s prediction, the network as a whole can get feedback on its prediction.

1.2.7 Neural Networks can Closely Approximate Many Important Functions

In their 1989 paper, Hornik et al. established that multilayer feedforward neural networks are "capable of approximating any measurable function to any desired degree of accuracy" (Hornik et al. (1989)). Furthermore, they state that "this implies that any lack of success in applications must arise from inadequate learning, insufficient numbers of hidden units or a lack of a deterministic relationship between input and target" (Hornik et al. (1989)). Therefore, at least in theory, neural networks have the potential to achieve high accuracy on any task which can be performed by a measurable function.

When the neural network is first created, it is initialized with random weights. These weights are where the neural network stores what it has learned. As a result, a newly created neural network with randomly initialized weights is unlikely to make good predictions. Thus, training a neural network can be thought of as gradually adjusting the network's weights so that it becomes a better and better approximation of the function that would correctly map the neural networks inputs to the desired outputs.

1.2.8 The Loss Function - How Good is the Network's Approximation?

To train a neural network to better approximate a desired function, it is necessary to determine how closely the neural network is currently approximating it. A simple way of doing this would be to count the number of times the network's prediction matched the corresponding label in the training data. However, this approach can be improved upon by instead calculating a loss function that measures how wrong the network was in its predictions.

Typically, loss functions will not only penalize the network for incorrect predictions, but will also penalize the network for being confident about incorrect predictions and for being unsure about correct predictions. This provides the network with better feedback, making training more effective. A change to the network's weights may not alter any of its predictions, however this change might make the network less confident about the predictions it got wrong, or more confident about the predictions that it got right.

For example, when training the neural cooking robot discussed previously, it might incorrectly classify a food with excessively high salt levels as delicious. When the neural network's weights are adjusted to help avoid this misclassification in the future, it is possible that its weights will not be adjusted enough to alter the network's classification prediction from delicious to not delicious. However, this change might make the robot less confident in its prediction that the food is delicious. Additionally, this change might make the network more confident in its predictions for some of the foods that it correctly classified. Both of these results are good. A neural network that predicts the wrong output for a given input but has a very low confidence in this prediction would be more likely to make the correct prediction for a slightly different input than another network that had very high confidence in its incorrect prediction. Furthermore, many applications using neural networks reveal the network's confidence level to their users. Since many users rely on a network's confidence level, including the network's confidence in its predictions into its loss function to incentivize it to have higher confidence levels correspond to higher probabilities of a correct prediction makes the network more useful to its users. For these reasons, loss functions typically incentivize a neural network to have high confidence levels in its prediction correspond to high probabilities of a correct prediction.

1.2.9 Updating the Network's Weights Based on the Loss Function

With the network's prediction and loss function in hand, the mathematical process of backpropagation is used to calculate how the neurons' weights need to be adjusted, so that the next time the network receives an input like this one it will be more likely to predict the input's label correctly. To do this, the slope of the loss function is calculated for the given input with respect to each of the network's weights. These slopes indicate how each of the weights should be increased or decreased to lower the loss function's value. Since the loss function is a measure of how different the neural network is from the ideal mathematical function that would perfectly perform this particular task, lower values of the loss function should result in a more accurate approximation of this ideal function.

1.2.10 Gradient Descent

The process of continually adjusting the weights of the neurons so that the network more closely approximates the theoretical ideal function is known as gradient descent. This is because if the loss function is graphed then it will look like the network is descending a gradient as it trains. For example, Figure 1.5 shows how gradient descent

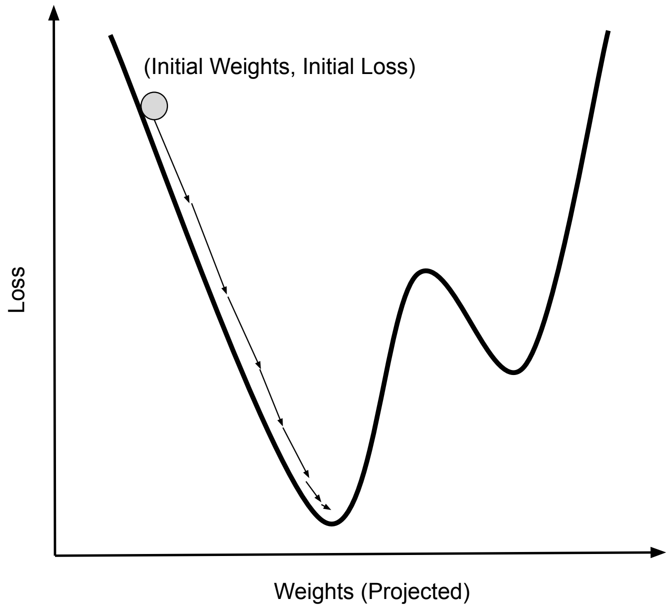

Figure 1.5: A visualization of gradient descent

might minimize a network's loss function by continually updating its weights.

1.2.11 Learning Rate

Now that backpropagation has calculated which direction each of the network's weights should be adjusted to reach a lower point on the loss function, the network's accuracy can be improved by making a small adjustment to each of its weights. The size of these small adjustments is largely determined by a number called the learning rate. A large learning rate makes it more likely that the network will over adjust and miss out on good solutions. Too small of a learning rate and the network will take too long to train. Additionally, too small of a learning rate makes it more likely that a network will get stuck in a local minima of the loss function. For an example of a learning rate that is too high, see Figure 1.6. In contrast, Figure 1.7 shows a learning rate that is too low. Figure 1.5 shows a good learning rate. Since the value of the best learning rate depends on the task the network is learning and on how long the network has been training, a number of optimizers have been created to adjust the learning rate throughout training. For example, the learning rate might start out high and then get smaller and smaller as training progresses.

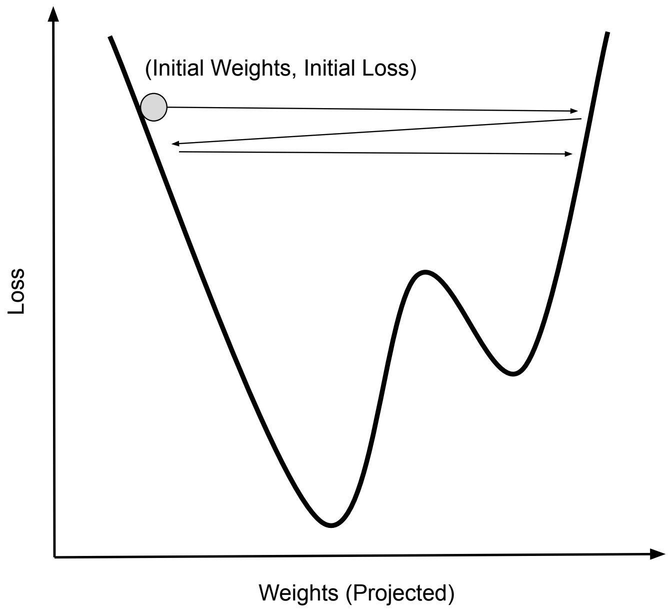

Figure 1.6: A visualization of gradient descent with too high of a learning rate

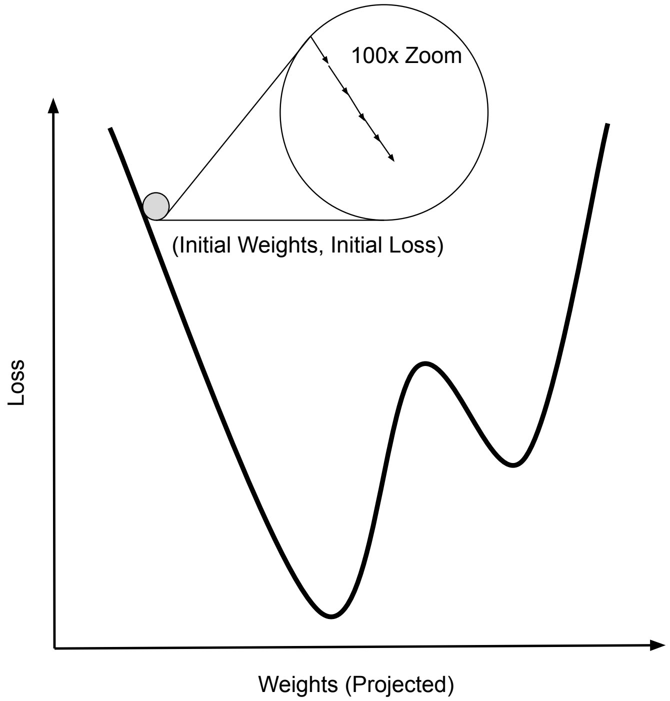

Figure 1.7: A visualization of gradient descent with too low of a learning rate

1.2.12 How Long Should a Neural Network Train?

The training duration is another important consideration when training a neural network. If the network is not trained for long enough, it will generally underfit the training data. However, if the network is trained for too long it will generally overfit the training data. Underfitting means that the network has not learned all that it can from the training data, while overfitting means that the network has started to memorize unimportant information about the training data. Figures 1.8, 1.9, 1.10 show a network's predictions underfitting, robustly fitting, and overfitting, respectively.

Another factor that influences whether a network will overfit is the ratio of the number of adjustable parameters the network has to the number of training examples. When a neural network has substantially more parameters than training examples it will be more likely to overfit the data.

One way of avoiding overfitting is to use a validation data set to determine when to stop training the neural network. A validation data set is typically constructed by separating a subset of the training data set from the rest. The neural network is then trained only on the training data that was not placed into the validation data set. This allows the validation data set to be used to evaluate the accuracy of the neural network during training. Additionally, it can give an idea of when the neural network is beginning to overfit the training data. As training continues, the loss on the training set will continually decline. In contrast, the loss on the validation set which evaluates the network but which it does not get to learn from, will generally decline for a while and then start to rise again when the network has started to overfit the training data (Goodfellow et al., 2016, 242). As a general rule, it is good to stop training when the validation loss is at its minimum value (Goodfellow et al., 2016, 242).

Finally, a test set is used to evaluate the neural network at the end. However, the test set should not be used to make any decisions about how to train the network or tune the hyper-parameters. If the test set was used to inform any of these decisions, then the neural network's accuracy on the test set may no longer represent the network's performance on different, real-world data.

1.2.13 Fine-Tuning Pre-Trained Neural Networks

A key problem with training state-of-the-art neural networks is that they require massive amounts of data and compute time to train. For some tasks such as causal

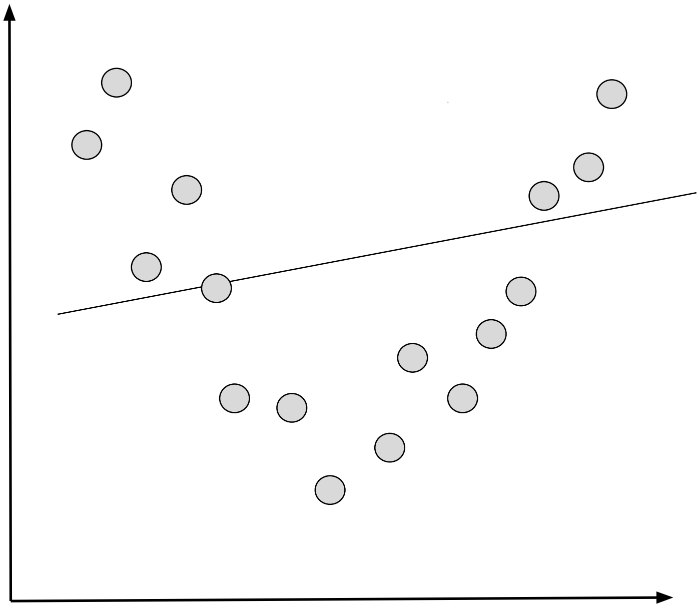

Figure 1.8: A visualization of underfitting

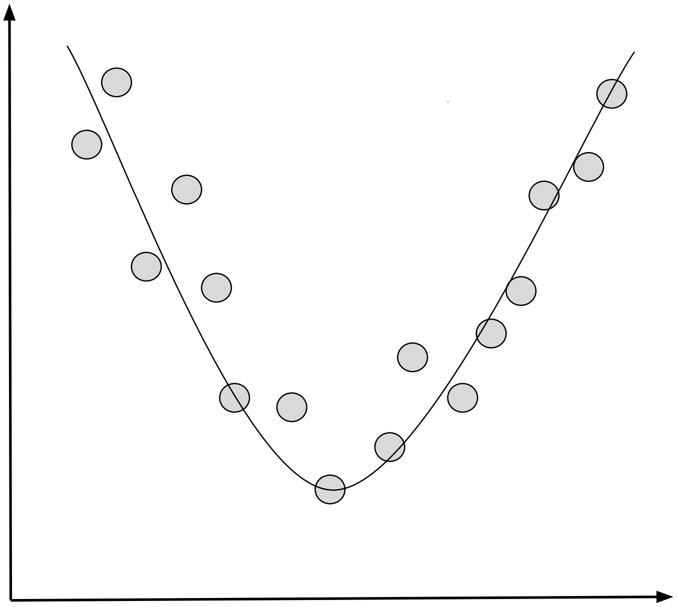

Figure 1.9: A visualization of fitting well

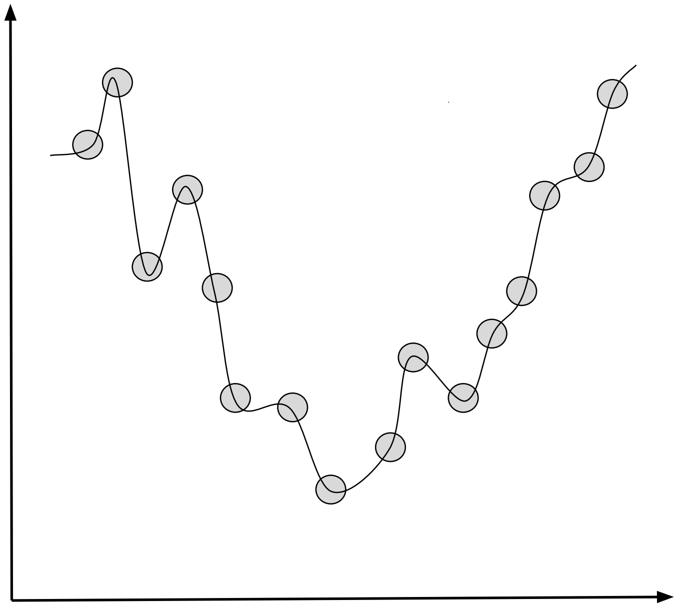

Figure 1.10: A visualization of overfitting

relation extraction, there is a very limited amount of labeled training data available. Furthermore, creating a large high-quality data set is very expensive. Part of the reason why so much training data is necessary to train a neural network is because randomly initialized neural networks start off with no baseline ”knowledge”. In contrast, when training a human to perform a task they are able to apply their related knowledge to the task. Recognizing this, researchers realized that one way of reducing the amount of necessary task-specific training data is to first train a neural network on a different task that will enable the neural network to acquire some baseline ”knowledge”.

In the context of NLP, the masked language model task has proven useful (Devlin et al. (2019)). In this task, the network is given a sentence from its training set with some of the words replaced with the token “[MASK]”. The network then needs to predict which word the mask token has replaced. The main benefit of this task is that nearly all of the text on the web can be used to train a network performing this task. It might appear that training a neural network on a task like masked word prediction is useless. It is true that there are relatively few useful applications for a network that can perform this task. However, the reason this task is used is that it is a difficult task that requires a good ”understanding” of natural language to perform well and that there is plentiful training data available. Thus, a network pre-trained to perform this task well will acquire a good general “understanding” of language and can then be trained again on a different task for which there is far less training data available (Devlin et al. (2019); Radford et al. (2019)).

Training a network that has been pre-trained on a large data set to perform a new task is called fine-tuning. Fine-tuning allows the neural network to transfer the general ”understanding” of language it learned on the pre-training task to the actual task that researchers want it to perform. This substantially reduces the amount of task-specific training data needed to get a particular level of performance. Furthermore, fine-tuning substantially reduces the amount of compute needed to train a task-specific neural network. This is likely because a pre-trained neural network's weights start much closer to the task's optimal weights than a randomly initialized neural network's weights. In this thesis investigation, the pre-trained neural network ROBERTA is fine-tuned on three different tasks that collectively perform causal relation extraction.

1.2.14 What is RoBERTa?

RoBERTa, which stands for Robustly Optimized BERT Pretraining Approach is a newer version of a previous state of the art neural network known as BERT (Liu et al. (2019)). BERT is a large neural network that was pre-trained on the masked language model task (Devlin et al. (2019)). The main difference between RoBERTa and BERT is that RoBERTa was trained on more training data and for longer (Liu et al. (2019)). Both RoBERTa and BERT were trained on BOOKCORPUS and the English WIKIPEDIA summing to about 16GB of training data (Liu et al. (2019)). However, RoBERTa was additionally trained on CC-NEWS (76GB), OPENWEBTEXT (38GB), and STORIES (31 GB). In addition to training on more data, RoBERTa was trained with different hyper-parameter values such as for the learning rate (Liu et al. (2019)). Despite relatively minor changes between BERT and RoBERTA, RoBERTa achieves better performance than BERT on almost all tasks (Liu et al. (2019)). Thus, RoBERTa appears to be the stronger pre-trained model and is fine-tuned in this thesis investigation.

1.3 SemEval 2010 Task 8 Data Set

1.3.1 What is the SemEval 2010 Task 8 Data Set?

SemEval is an international natural language processing group on a mission to advance the state of the art in semantic analysis. To this end, they continually create high quality annotated data sets for progressively more challenging problems related to understanding the semantics of natural language. The SemEval 2010 Task 8 data set is of particular interest to this thesis investigation. The task is titled "Multi-Way Classification of Semantic Relations Between Pairs of Nominals". Tables 1.1 and 1.2 show each of the semantic relations in the data set and their frequency. Overall, the data set contains 10,717 sentences in which a pair of nominals is labeled along with the relation between them and the direction of the relation. Of these 10,717 labeled sentences, 8,000 are set aside for training. The other 2,717 labeled sentences are stored in a separate file and are to be used for evaluating the performance of an AI system trained on the first 8,000 sentences. A sentence along with its labels will be referred to as a data-point. Example 1 shows the base formatting of the data set.

Table 1.1: SemEval 2010 Task 8 Train Set Relation Distribution (Hendrickx et al. (2010b))

Relation Type	Number of Occurrences	Percentage
Other	1410	17.63%
Cause-Effect	1003	12.54%
Component-Whole	941	11.76%
Entity-Destination	845	10.56%
Product-Producer	717	8.96%
Entity-Origin	716	8.95%
Member-Collection	690	8.63%
Message-Topic	634	7.92%
Content-Container	540	6.75%
Instrument-Agency	504	6.30%
Overall Sum	8000	100%

Table 1.2: SemEval 2010 Task 8 Test Set Relation Distribution (Hendrickx et al. (2010b))

Relation Type	Number of Occurrences	Percentage
Other	454	16.71%
Cause-Effect	328	12.07%
Component-Whole	312	11.48%
Entity-Destination	292	10.75%
Message-Topic	261	9.61%
Entity-Origin	258	9.50%
Member-Collection	233	8.58%
Product-Producer	231	8.50%
Content-Container	192	7.07%
Instrument-Agency	156	5.74%
Overall Sum	2717	100%

Example 1
7 ”The current view is that the chronic <e1>inflammation</e1>in the distal part of the stomach caused by Helicobacter pylori <e2>infection</e2>results in an increased acid production from the non-infected upper corpus region of the stomach.”
Cause-Effect(e2,e1)
Comment:

The two items that the semantic relation relates are labeled with <e1>and <e2>tags. These tags will simply be referred to as e1, e2 tags or labels from here on. The relation type is contained below the sentence and the ordering of e1 and e2 after the relation type label describes the direction of the relation occurring between the two nouns. Thus, in the example above the labels state that “infection” is causally related to “inflammation” and that they are the cause and the effect, respectively.

1.3.2 How Representative is SemEval 2010 Task 8?

While the data set provides a fairly good representation of how causal relations are expressed in natural language texts, their annotation guidelines explain some key differences between their data set and the way these may be expressed in natural language texts. For example, SemEval 2010 Task 8 considers ”only instances of semantic relations pertaining to situations in the real world and [excludes] instances [semantic relations] as pertaining to situations in some other world defined by counterfactual constraints elsewhere in the context" (Hendrickx et al. (2010a)). As such, instances of semantic relations that do not pertain to situations in the real world have been excluded from the data set (Hendrickx et al. (2010a)). Notably, conditional clauses (if, unless, assuming that...) and imperative clauses (Have fun, Enjoy your meal...) have been excluded from the data set (Hendrickx et al. (2010a)). Furthermore, sentences containing anaphoric expressions (those that rely on external context to be understood) are also excluded from the data set (Hendrickx et al. (2010a)).

Beyond the real world constraint, SemEval 2010 Task 8 also imposes restrictions on which nominal expressions can be marked (Hendrickx et al. (2010a)). For example, "only base noun phrases whose head is a common noun" can be marked as either e1 or e2" (Hendrickx et al. (2010a)). "A base noun phrase is a noun and its premodifier's (e.g., nouns, adjectives, determiners)" (Hendrickx et al. (2010a)). For example, "lawn mower" is a base noun phrase (Hendrickx et al. (2010a)). In contrast, "the engine of the lawn mower" is a complex noun phrase and would not be marked in the data set (Hendrickx et al. (2010a)). Given the noun phrase "a brown dwarf star", "there are five segments which have the structure of base noun phrases" (dwarf, star, dwarf star, brown dwarf, brown dwarf star) (Hendrickx et al. (2010a)). Furthermore, in the data set the entities e1 and e2 will typically each be only a single word long (Hendrickx et al. (2010a)). They can only contain multiple words when the multiple words make up a lexicalized noun phrase (Hendrickx et al. (2010a)). A lexicalized noun phrase is a noun phrase that acts as a single word (Godby, 2002, iii). An example of a lexicalized noun phrase is "high school" (Godby, 2002, iii). Here, the words "high school" are both necessary to evoke the intended meaning (Godby, 2002, iii). Another example of a lexicalized noun phrase is "garbage man" (Godby, 2002, iii).

Another restriction placed on the assignment of relations to sentences is that the relation must still hold at present. For example, "even though 'The <e1>ball</e1>is retrieved from the <2>hole</e2>.' might evoke both Content-Container and Entity-Origin, Content-Container is out because the ball has been removed from the hole." (Hendrickx et al. (2010a)). Furthermore, when a sentence expresses multiple relations a rough guideline based on "informativity order" is used to determine which relation is assigned to the sentence (Hendrickx et al. (2010a)). The order is "Entity-Origin / Entity-Destination<Message-Topic<Instrument-Agency<Content-Container<Component-Whole / Member-Collection<Cause-Effect / Product-Producer" (Hendrickx et al. (2010a)). Thus, a sentence that could be labeled as either Product-Producer or Entity-Origin will be labeled Entity-Origin (Hendrickx et al. (2010a)).

1.3.3 How Does SemEval 2010 Task 8 Define a Causal Relation?

Although the SemEval 2010 Task 8 data set contains labels for 10 different types of semantic relationships, the cause-effect relationships are of particular interest in the present investigation. Cause-effect relationships make up 12.54% of the training data set file. Thus, of the 8,000 labeled sentences in the training data set file 1,003 are labeled as containing a causal relation. SemEval 2010 Task 8 uses the following definition of a cause-effect relationship (Hendrickx et al. (2010b)).

Cause-Effect(X, Y) is true for a sentence S that mentions entities X and Y if and only if:
(1) S, X and Y are in accordance with the general annotation guidelines
(2) the situation described in S entails that X is the cause of Y,
or that X causes/makes/produces/emits/... Y.

Although causal relations are expressed and interpreted in a wide variety of ways, the more limited sense in which we are interpreting causal relations in the present investigation is useful for AI systems because it places a higher burden on what kind of information can count as causal information. Since this information would then be used to construct a knowledge base for the machine to reason from it is beneficial to require a high level of confidence that the sentence is intended to communicate a causal relationship. This will reduce the number of reasoning errors a model would make by reducing the number of false premises that it would attempt to reason from.

1.4 Evaluation Metrics

Evaluation Metric formulas:

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Note that in the formulas above tp, fp, and fn, are short for true positives, false positives, and false negatives respectively.

When evaluating the performance of a machine learning system there are many different performance metrics that can be used. One of the simplest metrics is the accuracy. The accuracy is the number of correct predictions divided by the total number of predictions. For some applications, accuracy is a sufficient evaluation metric. However, it is often better to use other evaluation metrics. Two other types of evaluation metrics are precision and recall. For a given label, a system's precision is the number of times the model correctly predicted the label divided by the number of times that the model predicted that label (Goodfellow et al., 2016, 418). In contrast, for a given label, a system's recall is the number of correct predictions for that label divided by the number of true occurrences of that label in the data (Goodfellow et al., 2016, 418). It is advantageous to use precision and recall when there are different costs associated with different types of predictions (Goodfellow et al., 2016, 418). Another metric that is commonly used in machine learning is the F1 score (Goodfellow et al., 2016, 419). It is a useful way to summarize the performance of a model with a single number (Goodfellow et al., 2016, 419).

Chapter 2

Causal Relation Extraction: a Three Step Process

2.1 Task Description

In this chapter, we will create a three step process for extracting causal relations with neural networks. The first step is to use a classifier neural network to identify sentences that contain a cause-effect relationship. The second step is to identify pairs of nouns that are causally related within these sentences. Finally, the third step is to determine the direction of the cause-effect relationship between each pair of nouns. Together these three steps can be used to extract cause-effect relationships from natural language texts.

2.2 Limitations and Simplifying Assumptions

Only causal relations between a pair of nouns will be extracted. Empirically this seems to be a large chunk of the causal relations occurring in natural language texts. A greater limitation is that typically only one word is extracted for the cause and the effect each. However, in many sentences, a noun phrase better identifies a cause and an effect than a single noun. In sentences where there are multiple causal relations, the three step process should be possible to extract each of them. However, it is not possible to extract causal relations occurring between sentences with this system. Furthermore, the system only attempts to extract explicit causal relations. In contrast, implicit causal relations are when there is only implied causality. For example in the sentences “The lit match was placed in the oil. The oil ignited.” The system would not try to extract the implied causal relation between placing the lit match on the oil and the oil igniting. There are three main reasons for this. First, it is much more difficult to accurately extract implied causal relations. Second, since there are large amounts of text available to use for causal relation extraction, it makes sense to trade recall for higher precision. Third, there are few if any data sets available with labeled implicit causal relations. Additionally, the restriction that the SemEval data set does not consider hypothetical scenarios means that many causal relations are not extracted. For example, causal relations occurring in if-then sentences are not extracted.

2.3 Architectural and Hyper-Parameter Choices

2.3.1 Which Pre-Trained Model?

Preliminary experiments were conducted using BERT, ALBERT, and RoBERTa on the SemEval 2010 task 8 data set. Additionally, preliminary experiments were conducted comparing using the large and the base model sizes. Overall, RoBERTa large resulted in the best performance in the preliminary experiments. Thus, RoBERTa large was selected to perform the each of the three steps involved in extracting causal relations in this thesis.

2.3.2 What Batch Size and what Input Length?

A batch size of 32 was used for all training rounds of all models. This batch size was selected because it is large enough that training proceeded quickly and that the model converged well but also small enough so as not to require too much VRAM. Additionally, after investigating the data set and determining that even the longest sentence could be represented in fewer than 128 tokens the model was modified to use an input length of 128 tokens only. This sped up the training speed by a factor of 4 and dramatically reduced the VRAM required to train the model.

2.3.3 How many Epochs?

The number of epochs that the model was trained for was determined by first training the model on a validation set that is a subset of the data originally set aside by SemEval for training. At first, the model was trained for a fairly large number of epochs. Then the model's loss on the validation set in each epoch was analysed to determine how many epochs to train for. The number of training epochs was selected by finding the last epoch in which the validation loss decreased. This strategy is known as early stopping (Goodfellow et al., 2016, 246). This is a commonly used regularization technique for deep learning and is frequently used to help prevent overfitting (Goodfellow et al., 2016, 246). ”Its popularity is due both to its effectiveness and its simplicity” (Goodfellow et al., 2016, 246).”

2.4 Step 1: Distinguishing between Causal and Non-causal sentences

2.4.1 Task Description

In the first step, the task is to classify sentences as either containing or not containing a causal relation. For example, a sentence like “The food is delicious!” should be classified as not containing a causal relation. In contrast, a sentence like “the explosion caused the loud sound” should be classified as containing a causal relation. An example of a more challenging non-causal sentence is ”Speculations on the causes behind the below capacity production of ethanol by the sugar factories in the state figured in the Legislative Council on Tuesday”. This sentence would likely fool a naive classifier because it contains the word “causes”, which is usually highly indicative of the presence of a cause-effect relationship. Notably, sentences that contain multiple cause-effect relationships should also be classified as causal

2.4.2 Task Motivation

The primary motivation for distinguishing between causal and non-causal sentences in step one is to save computing power. It would be possible to identify sentences containing causal relations by checking if each pair of nouns occurring in a sentence are causally related or not. However, some longer sentences have 100 or more possible unique noun pairings. Furthermore, sentences expressing causal relations appear to occur relatively infrequently. In the SemEval 2010 task 8 data set, they make up about 12.5 percent of sentences (Hendrickx et al. (2010b)). The percentage of sentences containing causal relations in the real world is not known. However, based on experience, it seems to be significantly less than 50%. Thus, since sentences containing causal relations are relatively uncommon, first distinguishing between causal and non-causal sentences to reduce the number of sentences that undergo the more computationally expensive second step of identifying which pairs of nouns are causally related should make the causal relation extraction process more efficient.

2.4.3 Constructing a Data Set

In the SemEval 2010 task 8 data set, each sentence contains one pair of e1 e2 tags that mark a pair of nouns (Hendrickx et al. (2010b)). For example, the sentence "The <e1>radiation</e1>from the atomic <e2>bomb explosion</e2>is a typical acute radiation." Uses e1 e2 tags to indicate that the relevant noun pair is "radiation" and "bomb explosion". To prepare the data to train a classifier neural network to perform this task, the e1 and e2 tags are removed. Since this task attempts to distinguish between sentences that contain no causal relations and those that do, the noun pair is not relevant. Furthermore, keeping the e1 and e2 tags could harm the classifier's generalizability.

Next, each of the sentence's relation types is relabeled. In the original data set, there are 10 different types of relations. However, for this task, it is only relevant if the sentence is labeled as causal or non-causal. Thus, all non-causal relations have the value of their label replaced with the same value. Note that this results in an imbalance between the size of the classes that the classifier is distinguishing between.

2.4.4 Why this Approach is not Comparable with other Papers

The results on this step are not comparable with other researcher's results on SemEval 2010 task 8 for two reasons. First, this task focuses on identifying whether a sentence contains a causal relation at all, whereas papers focusing on identifying causal relations in the SemEval 2010 task 8 data set classify when a particular pair of marked words are causally related. Second, the markers around the specified words were removed so researchers working on the other task would be using slightly different input data.

2.4.5 Data Augmentation with Back Translation - a Negative Result

Data augmentation is the process of creating new training data by creating slightly modified copies of the original training data. Data augmentation is widely used for image processing tasks because it is very straightforward and generally effective.

Common image data augmentation techniques include cropping, resizing, and mirroring the images. However in natural language processing tasks, data augmentation is less frequently used and more tricky. One common data augmentation technique for natural language data sets known as back translation involves translating the data into another language and then translating it back into the original language. The hope is that the translation will preserve the important properties of the data while altering the wording.

In this thesis investigation an attempt was made to improve classification accuracy by using back translation to augment the data set for the classifier that detects if a sentence contains a causal relation or not. Back translation appears to preserve this property of a sentence. However, back translation was not attempted on the data sets for steps two and three of the three-step causal relation extraction system proposed in this investigation. The properties important for these tasks were not preserved well enough by back translation for the augmentation strategy to be useful.

Another important consideration when applying back translation is which pair of languages to translate between. Since a better translation is more likely to preserve the important properties of text, back translation is only effective when good translations are available. Languages that are frequently translated between tend to have better translations. For this reason, it is best to use commonly translated language pairs like English and Spanish, or English and Chinese. Another important factor for back translation is how different the two languages are from each other. Translating between languages that are less similar tends to produce greater variability in wording. This greater variability should allow the model to better generalize to unseen data. For this reason, English and Chinese were selected as the translation language pair instead of English and Spanish. While the English-Spanish back translations frequently preserved the meaning of the sentences, there were only minimal differences between the original and back translated text.

In this investigation, Google Cloud Platform's Translation API performed the data translations. Originally, each data-point had its sentence and its corresponding labels translated separately. However, this resulted in bad translations due to a lack of context. For example, when asked to translate the Spanish word "configuración", answering "setting" is reasonable. However, in the context of a particular sentence this may no longer be a good translation. Consider the following example.
Original:
"The system described above has its greatest application in the array configuration of antenna elements."

e1 contents: ”configuration”
e2 contents: ”element”

Spanish:
”El sistema descrito anteriormente tiene su mayor aplicación en la configuración de matriz de elementos de antena”
e1 contents: configuración
e2 contents: elemento

Individually Translated English:
”The above system has its biggest application in the array configuration of antenna elements.”
e1 contents: setting
e2 contents: element

Translated English:
”The system described above has its greatest application in the array configuration of antenna elements.”
e1 contents: configuration
e2 contents: ”element”

In the example above, the word configuration is incorrectly translated differently in the sentence and in the label due to a lack of context. To address this problem, a sentence and its labels are translated simultaneously. Another issue with translating the text is that sometimes correct but different translations are given for a sentence and its labels. However, it is required that the labels for the e1 and e2 contents be a sub-string of the sentence. This problem can be solved by replacing the contents of the label that does not occur in the sentence with the most similar word that does occur in the sentence.

One good way to evaluate the similarity of two words is to convert the words into vectors and to then take the cosine similarity of each pair of words. The paper *Efficient Estimation of Word Representations in Vector Space* demonstrated that a word’s vector representation encodes many of the semantic and syntactic relationships between it and other words (Mikolov et al. (2013)). For example, two words with similar meanings such as “lime” and “lemon” will have more similar vector rep-

resentations than more distantly related words like “rocket” and “broccoli”. One of the ways this similarity is represented by the vectors is by having each of the vectors point in a similar direction. A well known way of measuring this similarity is to calculate the cosine similarity of each of the two vectors. The cosine similarity is defined to be the cosine of the angle between the two vectors. Thus, the cosine similarity of two vectors ranges from a value of 1 for two vectors pointing in exactly the same direction to a value of -1 for two vectors pointing in exactly the opposite direction. Below is the cosine similarity of the word pairs mentioned above.
“lime” and “lemon”: 0.5828212
“rocket” and “broccoli”: 0.13929227

By using the Word2Vec vectors and cosine similarity, a large number of datapoints broken in translation are repaired. The mismatching translation can be corrected by calculating the cosine similarity of the mismatched label with every word in the sentence and replacing the mismatched label with the most similar word from the sentence. In this investigation, this process is referred to as validation. Although this technique corrects the majority of mismatches, sometimes the label will be mismatched and will contain multiple words. It is more difficult to correct these mismatches because Word2Vec only provides a vector for individual words. Thus, these more difficult mismatches are simply removed from the data set. Example 2 demonstrates one of these more difficult mismatches. However, it should be noted that a similar technique could be used to handle these more difficult mismatched translations as well. Example 3 demonstrates how a mismatched label can be corrected.

Example 2: Original
”People have been moving back into downtown.”
e1 contents: People
e2 contents: downtown

After Translation:
”People have moved back to the city.”
e1 contents: People
e2 contents: urban area

Example 3:
Original:
”The waste, a mixture of gasoline, water and caustic soda, gave off toxic fumes.”
e1 contents: waste
e2 contents: fumes

After Translation:
”The mixture of waste, gasoline, water and caustic soda releases toxic fumes.”
e1 contents: waste
e2 contents: smoke

After validation:
”The mixture of waste, gasoline, water and caustic soda releases toxic fumes.”
e1 contents: waste
e2 contents: fumes

After an augmented data set was created with back translation, preliminary experiments were conducted to determine how effective it was. It was found to modestly improve performance of the bert-base-uncased model when 20% of the translated data was included in the training set. However, training bert-large-uncased on this data set showed no improvement in performance. Furthermore, when using more of the translated data performance declined. Thus, the augmented data was not used to train any of the neural networks evaluated on the test set.

2.5 Step 2: Identifying the Candidate Cause and the Candidate Effect

2.5.1 Task Description

In the second step, the classifier needs to identify all the pairs of nouns occurring in the sentence that are causally related. This is done in two steps. First, the nouns are identified and paired up. Next, the neural network classifies each pair as either being causally related or not causally related. If the pair is found to be causally related then they are called the candidate cause and the candidate effect. Note that the input to the second step is a sentence that is assumed to contain a causal relationship, since only sentences that the first step identified as containing a causal relation are fed into the second step. An example of a possible input to the second step is the sentence “A virus infecting a nose can cause inflammation.” This sentence contains the nouns “virus”, “nose”, and “inflammation”. Thus, the possible noun pairings are (virus, nose), (virus, inflammation), and (nose, inflammation). For each of these noun pairings, a version of the sentence is generated with the pair of nouns replaced with the tokens 0001 and 0002 respectively. For example, using the first pairing the resulting sentence would be “A 0001 infecting a 0002 can cause inflammation”. This sentence would then be provided to the classifier neural network which would (hopefully) determine that the two words are not causally related in this sentence.

2.5.2 Identifying and Pairing Nouns

To generate the pairings of nouns for a sentence, the spaCy library is used to identify the part of speech of all of the words in the sentence. From here all the nouns are simply paired up. It should be noted that although the spaCy en_core_web_lg model has very high accuracy at identifying the part of speech it is not perfectly accurate. For example, in some sentences where a gerund is used as a noun, spaCy will classify it as a verb. For the data in the training set, it was determined that roughly 95% of the time a list of all possible pairings of nouns contained the pair of nouns that were labeled as being causally related. This means that about 5% of the time Spacey mislabeled the part of speech of at least one of the two nouns that are a member of the labeled causal relation.

Since the number of possible pairings of nouns in a sentence is very large, heuristic methods were used to greatly reduce it. The first heuristic that was used was eliminating all possible pairings where the two nouns in question were not separated by a verb. The thinking was that two nouns that are right next to each other with no verb between them would rarely if ever express a causal relation. However, this heuristic turned out to only have a recall of about 75%. An example of a sentence that is not handled correctly by this heuristic is ”The man got cancer from smoking.” In this case, cancer is the effect and smoking is the cause. Furthermore, there is no verb separating cancer and smoking because the word “from” is being used to express the causal relationship between these two words instead, and the word ”from” is a preposition, not a verb. A better heuristic was created by using the spaCy dependency labels for the sentence. The spaCy dependency labels are attached in Table 2.1. The heuristic was first manually created by only allowing noun pairings where the dependency type of each of the nouns was either a subject or an object. This heuristic had better recall than the verb separation heuristic explained previously. To further optimize the heuristic an automated system was created to test how adding each dependency tag into the list of allowed or dependency tags would affect the recall. Using this system a more complicated heuristic was created that strikes a balance between eliminating a large number of noun pairings well retaining as many of the pairings that contain causal relationships as possible. The heuristic had a recall of about 85%, while eliminating half the noun pairings.

Depending on the systems user's available computing power and on the amount of data they have to extract cause-effect relationships from, it may or may not be desirable to use the heuristic to reduce the number of noun pairings being considered.

2.5.3 Constructing a Data Set

Now that we can generate pairings between nouns we need a way to classify whether or not there is a causal relationship between the two specified nouns. For this task, the e1 and e2 labels were originally added back in to communicate to the model which two words are being considered. However, after adding these tags back in the input tokenization length had noticeably increased. This is because the e1 e2 tags use sequences of characters that were very uncommon in RoBERTa's pre-training data and so each occurrence of an e1 or e2 open or close tag required three or four tokens from the model's vocabulary to represent. The tokenization length was reduced and the generalizability of the model was likely improved by replacing the e1 e2 tags and the contents contained between them with the numbers 0001 and 0002. These numbers were chosen because they are contained in RoBERTa's vocabulary. Thus, they can be represented with a single token. Furthermore, these numbers were likely infrequently used in the pre-training data and likely have little meaning to the pre-trained RoBERTa model. Furthermore, generalizability was likely improved because any two sentences that only differed by the nouns in the pair would now look identical to the model.

Moreover, this makes sense in a system that is trying to extract causal relations because if the occurrence of a causal relation depended on the particular words that were being used then the model would be more inclined to accept causal relations that it happened to learn in pre-training. Furthermore, humans can identify cause-

Table 2.1: SpaCy Dependency Labels

Dependency Label	Description
ROOT	None
acl	clausal modifier of noun (adjectival clause)
acomp	adjectival complement
advcl	adverbial clause modifier
advmod	adverbial modifier
agent	agent
amod	adjectival modifier
appos	appositional modifier
attr	attribute
aux	auxiliary
auxpass	auxiliary (passive)
case	case marking
cc	coordinating conjunction
ccomp	clausal complement
compound	compound
conj	conjunct
csubj	clausal subject
csubjpass	clausal subject (passive)
dative	dative
dep	unclassified dependent
det	determiner
dobj	direct object
expl	expletive
intj	interjection
mark	marker
meta	meta modifier
neg	negation modifier
nmod	modifier of nominal
npadvmod	noun phrase as adverbial modifier
nsubj	nominal subject
nsubjpass	nominal subject (passive)
nummod	numeric modifier
oprd	object predicate
parataxis	parataxis
pcomp	complement of preposition
pobj	object of preposition
poss	possession modifier
preconj	pre-correlative conjunction
predet	None
prep	prepositional modifier
prt	particle
punct	punctuation
quantmod	modifier of quantifier
relcl	relative clause modifier
xcomp	open clausal complement

effect relationships occurring in text regardless of what the particular nouns are. For example, given the sentence ”A causes B ” a human will have no difficulty realizing that there is a causal relation between A and B. Likewise, we would also have no difficulty with the sentence “Lambda causes gamma”. We might not know what lambda is or what gamma is but based on the structure of the sentence we can tell that a causal relation is being expressed between lambda and gamma such that lambda causes gamma. This modification slightly improved accuracy on the validation set. Another modification that was made to the original SemEval 2010 task 8 training data set was that non-causal sentences were excluded. At this point, step one will have already determined that the sentence contains a causal relation.

Additionally, since the original data set only contained noun pairings for which there is a causal relationship between the two specified nouns, the data set was augmented to include noun pairings between nouns that are not causally related. This was achieved by first generating noun pairings for all sentences that contained fewer than 15 nouns. Next, all pairings that were labeled in the original data set as being causally related were removed. Then all of the pairs were labeled as non-causal. Finally, the sentences that were originally labeled as containing causal relations were added back. The resulting data set contains all of the causal relations in the original data set and it also contains a large number of pairings between nouns occurring in these causal sentences that were not marked in the original data set as being in a causal relationship. In this new data set, 15.8% of the 6,294 pairings were causally related. In contrast, in the SemEval 2010 task 8 data set 100% of noun pairings in causal sentences were between causally related nouns.

Note that some of the sentences in this new data set are likely incorrectly labeled. This is because some sentences in the original data set contained multiple causal relations. For example, the sentence ”Eating salt and exposure to heat causes thirst” contains multiple causal relations. In this sentence, there are two causal relations. The first causal relation is that salt causes thirst and the second causal relation is that exposure to heat causes thirst. With this data creation process, only the noun pairs that were originally labeled as being causally related end up labeled as causally related.

One benefit of this data creation process is that it allows for the creation of significantly more data than was originally present in the data set. The original data set has only about 1,000 sentences that contained causal relations and it contained zero pairings of nouns that were labeled as not containing any relations between them. When using this method it was possible to create a much larger data set.

Furthermore, a even larger data set could have been constructed by removing the restriction that sentences with more than 15 nouns be excluded from the data set. These sentences were excluded for two reasons. First, these long sentences tend to be very complicated. For example, consider the sentence "The current view is that the chronic inflammation in the distal part of the stomach caused by Helicobacter pylori infection results in an increased acid production from the non-infected upper corpus region of the stomach" (Hendrickx et al. (2010b)). According to the data set, the cause is "infection" and the effect is "inflammation". Secondly including these sentences would dramatically increase the proportion of data-points in the data set that contained no causal relation. Choosing 15 for the number of nouns seems to be a good choice because it resulted in 15.8% of pairings being causal. These class sizes are not wildly imbalanced, preventing the model from attaining high accuracy by simply always guessing non-causal. Furthermore, although this data imbalance between the classes likely biased the classifier towards predicting non-causal this seems to be a good bias to have. In the real world, the majority of nouns in a sentence do not seem to be causally related. Note that limiting the data set to sentences containing 15 or fewer nouns likely decreases the difficulty of the task by removing some of the more difficult sentences.

2.5.4 Why this Approach is not Comparable with other Papers

The results for this step are not comparable with other researcher's results because this step classifies when a pair of nouns in a sentence containing a causal relation are causally related. This is a more difficult task than in SemEval 2010 task 8 because in the SemEval data set whenever a sentence contains a causal relation the nouns that are causally related are marked with the e1 e2 tags. Thus, the task performed by causal classifiers on the SemEval 2010 task 8 data set is more similar to step 1 than to this task. Additionally, in this task, the markers denoting which pair of nouns are being classified and the nouns themselves have been replaced with the numbers 0001 and 0002. This forces the classifier to decide if the two nouns are causally related based on the structure of the sentence instead of being able to use outside knowledge about causes and effects to make a decision. This likely improves the generalizability of the model because all sentences that only differ in the selected pair of nouns will appear identical to the classifier.

2.6 Step 3: Identifying the Direction of the Cause-Effect Relation

2.6.1 Task Description

For the third step, the classifier is tasked with predicting the direction of the cause-effect relationship between two nouns given that there is a cause-effect relationship between those two nouns. For example, given the sentence “the old cheese caused the foul smell”. The first step would (hopefully) identify the sentence as containing a causal relation. The second step would (hopefully) identify cheese and smell as causally related. Then the sentence would be modified like so “The old 0001 caused the foul 0002”. The step 3 classifier would then (hopefully) identify 0001 as causing 0002. At this point, all the necessary information is available for a basic script to extract and store a causal relation between “cheese” and “smell”. There are no known research papers performing this task in a comparable way so this step is not compared.

2.6.2 Constructing a Data Set

Originally this step was conducted by having the model predict the direction of the relationship occurring between the two nouns for each of the 10 different types of relations contained in the SemEval 2010 data set. Additionally, this task was originally conducted by placing e1 e2 labels around the nouns being considered. Originally using bert-large-uncased, this task had a validation accuracy of about 82%. However, this task like the second step was modified to replace the e1 e2 labels and their contents with 0001 and 0002. This change brought the validation accuracy up to around 88%. Furthermore examining the data more closely, it was determined that for some of the other relation types contained in the data set it was significantly more difficult to determine the direction of the relation than for others. Since at this point in the process the input sentence is known to contain a cause-effect relationship between the two specified nouns, a new data set was constructed to train for this task. The modified data set only contains sentences from the original training data set that were labeled as containing a cause-effect relationship. In this way, the classifier could “focus” on predicting the direction of cause-effect relationships. Although this reduced the amount of data available to train for this task by roughly a factor of 8. This reduction in the data makes sense given the task that we are trying to perform and is likely somewhat offset by the increased generalizability given by replacing the contents of the e1 and e2 labels with individual tokens. Additionally, this model was also switched from bert-large-uncased to roberta-large. After making all of these changes the accuracy for this task on the validation set increased from its original value of 82% to 98% accuracy.

2.6.3 Data Augmentation for Step 3 - a Negative Result

An additional form of data augmentation was attempted for this task. The idea was to switch the order of the e1 and e2 labels in the sentence and to also switch the direction of the relations label. This resulted in a data set twice as large and should teach the model to cope with situations where the e1 label does not always occur before the e2 label in the sentence. Interestingly the model does not seem to be able to learn how to perform this task. After an extended period of training the validation accuracy remained at 50%, the accuracy expected for random guessing. Thus, this form of data augmentation was not used to train the final classifier.

2.7 Overall Three Step Process

2.7.1 Results

Table 2.2: Overall Results

Step	Class	Overall Accuracy	Precision	Recall	F1 Score
Step 1	Causal	97%	84%	93%	89%
Step 2	Causal	93%	76%	81%	79%
Step 3	Cause First	97%	98%	94%	96%
Step 3	Effect First	97%	96%	98%	97%
Overall System	Causal	-	-	79%	-

Note that in the table above overall accuracy is the accuracy across all classes not for one specific class. Also note that step 3 had a much smaller evaluation set. Finally, note that the recall does not take into account unlabeled causal relations in the data set.

The results of the overall approach along with the results in each of the three steps are shown in the Table 2.2. The table reveals that step 1 had a very high overall accuracy but that it had a noticeably lower precision and recall for causal sentences. This suggests that the neural network performed worse on sentences that were causal than on those that were not. The data imbalance between the classes of causal and non-causal sentences may partially explain this result.

The table reveals that step 2 had a significantly lower performance across all four measurements. This likely reflects the greater difficulty of identifying causally related nouns in a sentence containing a causal relation than in simply identifying when a causal relation is present.

For step three, the model had high accuracy, precision, and recall, resulting in a high F1 score. The metrics are reported for both sentences where the cause occurs first and sentences where the effect occurs first because both of these classes are of equal interest to this thesis investigation. It should be noted that because step three can only be evaluated on sentences that contain a causal relation there was only 323 sentences available for evaluation. In contrast, step 2 had 2,159 sentences for evaluation and step 3 had 2,682 sentences for evaluation. Thus, the exact percentages for step 3 are less solid than for the other two steps.

Finally, the overall recall of the system was determined to be 79%. It makes sense that the systems performance overall is lower than for any of the individual steps because the overall system is only considered to have correctly extracted a causal relation if that particular relation is correctly handled by each of the three steps. Looking at the table, it seems likely that the low relatively low recall in step 2 is bringing down the recall of the overall system.

It was not possible to calculate the recall for all causal relations contained in the data set because not all of them are labeled. For example, if a sentence in the data set contains multiple causal relations then only one of them will be labeled. Thus, the recall displayed in Table 2.2 only takes into account causal relations labeled in the data set. For the same reason as why the overall system's reported recall is a rough estimate of the true recall, it was not possible to calculate the other metrics for the overall system.

Another important consideration is that the spaCy noun identifier system's inaccuracies are not reflected in the presented metrics. Thus, if this step was included the metrics presented for step 2 and the overall system would likely be noticeably lower. This is because if spaCy fails to identify a word as a noun when working on unlabeled sentences the system will be unable to extract it.

2.7.2 Why this Process is not Comparable with other Papers

The overall system is not comparable with other results. One reason for this is that the SemEval 2010 task 8 data set always correctly labels the most relevant noun pair. For sentences containing a causal relation in the SemEval 2010 task 8 data set this means that the problem of determining if the specified pair of nouns is causally related reduces to the problem of determining if the sentence contains a cause-effect relationship. In contrast, step 2 of the system described in this thesis was trained and tested on a data set where the majority of the time that the sentence contained a cause-effect relationship the specified noun pair was not causally related. This is expected to lower accuracy when compared to the SemEval data set. However, it is also a more realistic assumption about natural language texts in general. Thus, there is reason to believe that the system described in this thesis would generalize better than other approaches that focus more directly on attaining the best performance on the SemEval 2010 task 8 data set.

Another key difference between the system described in this thesis and in others is that for steps 2 and 3 of the causal relation extraction system the labeled nouns have been replaced with the tokens 0001 and 0002. This prevents the neural networks from using prior knowledge about which nouns cause which other nouns to inform their classification decisions. It is unclear if this change increases or decreases the difficulty of the classification tasks. The models are only able to make their decisions based on the structure of a sentence, but they receive their input in a format that is easier for them to work with. However, this change should increase the generalizability of the model. Finally, the inclusion of the first step in extracting causal relations may be decreasing the overall accuracy of the whole system. However, it should greatly reduce the amount of computing power required.

Conclusion

In this thesis, we created a three step process to extract causal relations with neural networks. The first step identified sentences that contained a causal relation. The second step distinguished between causally and non-causally related nouns. Finally, the third step determined the direction of the causal relation between the two nouns. All three steps were trained and evaluated on modified versions of the SemEval 2010 task 8 data set. The first step had an F1 score of 89%. Although the inclusion of this first step in the three step process likely reduces the overall systems effectiveness it can reduce the amount of computation resources needed to extract causal relations. The second step had an F1 score of 79%. This was much lower than for the other steps, likely reflecting the difficulty of identifying which pairs of nouns in a sentence containing at least one causal relation are causally related. The third step had an F1 score of 96% for sentences in which the cause occurred first and an F1 score of 97% for sentences in which the effect occurred first. These relatively high scores may be the result of this task being easier to perform than those in steps 1 and 2. Finally, the overall system had a measured recall of 79%. However, this measured recall does not reflect unlabeled causal relations in the data. Furthermore, the recall would likely be lower if the errors resulting from a failure to identify nouns by spaCy was taken into account. Overall, the results were fairly promising.

Further research could investigate alternative approaches for performing each of the three steps. At the time of writing, one possible approach appears particularly promising. The idea is to apply the technique described in the paper *Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference* (2020). The paper argues that one of the reasons large amounts of training data are required to train neural networks to perform natural language classification tasks that would be easy for humans is that the neural network has to learn how to perform the task without a task description (Schick & Schütze (2020)). The paper suggests reformatting a classification problem such as any of the ones in the three step process described in this thesis, so that the neural network will be prompted to answer a question by filling in a masked word (Schick & Schütze (2020). For example, the neural network might receive an input such as "A causes B. Does this sentence contain a causal relation? [MASK]". The neural network would then predict the word it considers to be the best replacement for the masked word. Some possible outputs include "Yes", "No", and "Definitely". If the replacement word for the mask indicates that there is a causal relation then that would be counted as the neural networks decision on the classification problem. In contrast, in the current system the neural network receives an input such as "A causes B" and needs to output either a 0 or a 1 to signify if the sentence contains a causal relation or not. It is never directly stated that it is supposed to output 0 whenever it detects a causal relation and that it is supposed to output 1 whenever it does not.

Appendix A

Causative Devices in the LOB Corpus

Causative Devices by Frequency

Device	Always Causal	Causal Occurrences
because	Yes	635
why	Yes	443
so	No	425
for	No	365
therefore	Yes	296
effect	Yes	278
cause	No	265
reason	Yes	212
thus	No	227
result	Yes	212
since	No	189
as	No	166
because of	Yes	142
so ... that	No	139
then	No	135
so that	No	127
due to	No	123
for (that) reason	Yes	93
lead to	No	83

from	No	81
hence	No	52
as a result of	Yes	48
bring	No	47
under	No	47
consequence	Yes	45
produce	No	45
consequently	Yes	44
result in	Yes	44
create	No	41
through	No	38
bring about	Yes	37
in view of	No	36
now that	No	32
arise from	No	29
owing to	No	26
arouse	No	26
present	No	25
such ... that	No	23
induce	No	22
result from	Yes	22
thanks to	Yes	21
in the light of	Yes	20
on account of	Yes	20
outcome	Yes	20
give rise to	Yes	19
give	No	18
inspire	No	17
on the grounds that	Yes	14
thereby	Yes	14
as a result	Yes	13
make for	No	11
contribute to	No	10
provoke	No	10
by reason of	Yes	8
derive from	No	8

prompt	No	8
what with	Yes	8
accordingly	Yes	7
by virtue of	Yes	7
generate	No	7
incur	No	7
raise	No	7
responsible for	No	7
as a matter of	No	6
in consequence	Yes	6
out of	No	6
source	No	6
arise out of	No	5
bring on	Yes	5
compel	No	5
engender	No	5
on the ground(s) of	Yes	5
on the strength of	Yes	5
with the result that	Yes	5
yield	No	5
evoke	No	4
excite	No	4
inasmuch as	Yes	4
in consequence of	Yes	4
motivate	No	4
occasion	Yes	4
pose	No	4
precipitate	No	4
produce of	No	4
root	No	4
rouse	No	4
stir	No	4
stir up	No	4
account for	No	3
as a consequence of	Yes	3
ascribe to	No	3

consequent on/upon	Yes	3
corollary	Yes	3
spark off	No	3
underlie	Yes	3
aftermath	No	2
as a consequence	Yes	2
attribute to	No	2
for reasons of	Yes	2
incite	Yes	2
in consideration of	Yes	2
on that account	Yes	2
on that score	Yes	2
put down to	No	2
seeing that	Yes	2
spring from	No	2
upshot	Yes	2
with the consequence that	Yes	2
at the bottom of	No	1
awaken	No	1
by consequence	No	1
consequential to	Yes	1
cos	Yes	1
emerge from	No	1
from reasons of	Yes	1
mainspring	Yes	1
spark	No	1
stem from	No	1
the whys and wherefores	Yes	1
beget	No	0
breed	No	0
by courtesy of	Yes	0
contributor to	No	0
foment	No	0
give occasion to	Yes	0
inspiration	No	0
proceed from	No	0

seeing as	Yes	0
spawn	No	0
spring out of	No	0

Devices with 0 occurrences are included despite not having any causal occurrences in the Lancaster-Oslo-Bergen (LOB) Corpus because they are still known to be causative devices. The column ”Always Causal” takes the value ”Yes” if every occurrence of the causative device in the LOB corpus was in a causal sentence. The column ”Causal Occurrences” is number of times that the device occurred in a causal sentence in the 1 million word corpus. It does not count the number of times that the device occurred in the corpus overall.

Data Source: Fang Xuelan & Kennedy (1992)

References

Asghar, N. (2016). Automatic Extraction of Causal Relations from Natural Language Texts: A Comprehensive Survey. *arXiv:1605.07895 [cs]*. ArXiv: 1605.07895. `http://arxiv.org/abs/1605.07895`

Brown, D. D. (2008). The Use of Causal Connections by Young Children: Implications for School Readiness. *NHSA Dialog*, *11*(1), 44–53. Publisher: Routledge _eprint: https://doi.org/10.1080/15240750701831909. `https://doi.org/10.1080/15240750701831909`

Cachola, I., Lo, K., Cohan, A., & Weld, D. S. (2020). TLDR: Extreme Summarization of Scientific Documents. *arXiv:2004.15011 [cs]*. ArXiv: 2004.15011. `http://arxiv.org/abs/2004.15011`

Cohen, A. D., Rosenman, S., & Goldberg, Y. (2021). Relation Classification as Two-way Span-Prediction. *arXiv:2010.04829 [cs]*. ArXiv: 2010.04829 version: 2. `http://arxiv.org/abs/2010.04829`

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805. `http://arxiv.org/abs/1810.04805`

Engelmore, R. S. (1987). Artificial Intelligence and Knowledge Based Systems: Origins, Methods and Opportunities for NDE. In D. O. Thompson, & D. E. Chimenti (Eds.), *Review of Progress in Quantitative Nondestructive Evaluation*, Review of Progress in Quantitative Nondestructive Evaluation, (pp. 1–20). Boston, MA: Springer US.

Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical Neural Story Generation. *arXiv:1805.04833 [cs]*. ArXiv: 1805.04833. `http://arxiv.org/abs/1805.04833`

Fang Xuelan, & Kennedy, G. (1992). Expressing Causation in Written English. *RELC Journal*, *23*(1), 62–80. http://journals.sagepub.com/doi/10.1177/003368829202300105

Godby, C. J. (2002). *School of The Ohio State University*. Ph.D. thesis, The Graduate School of The Ohio State University.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Group, S. M. (2010). Causal Patterns in Science. http://causalpatterns.org/causal/causal_types.php

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*. ArXiv: 1512.03385. http://arxiv.org/abs/1512.03385

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2010a). Data Creation Guidelines. https://docs.google.com/document/d/1Et-82axjZURSGE9a9znkJoMzEuLFIILohceMlZ1_1Fc/preview?usp=embed_facebook

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2010b). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. *arXiv:1911.10422 [cs]*. ArXiv: 1911.10422. http://arxiv.org/abs/1911.10422

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://www.sciencedirect.com/science/article/pii/0893608089900208

Johansson, S. (1978). Lancaster-oslo-bergen corpus of modern english (LOB) : [tagged, horizontal format] / stig johansson. Oxford Text Archive. http://hdl.handle.net/20.500.12024/0167

Kendeou, P., van den Broek, P., White, M., & Lynch, J. (2009). Predicting Reading Comprehension in Early Elementary School: The Independent Contributions of Oral Language and Decoding Skills. *Journal of Educational Psychology*, *101*, 765–778.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv:1909.11942 [cs]*. ArXiv: 1909.11942. http://arxiv.org/abs/1909.11942

Lefkowitz, M. (2019). Professor's perceptron paved the way for AI – 60 years too soon. https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692. http://arxiv.org/abs/1907.11692

Lorch, E. P., Diener, M. B., Sanchez, R. P., Milich, R., Welsh, R., & van den Broek, P. (1999a). The effects of story structure on the recall of stories in children with attention deficit hyperactivity disorder. *Journal of Educational Psychology*, *91*(2), 273–283. http://doi.apa.org/getdoi.cfm?doi=10.1037/0022-0663.91.2.273

Lorch, E. P., Sanchez, R. P., van den Broek, P., Milich, R., Murphy, E. L., Lorch Jr., R. F., & Welsh, R. (1999b). The Relation of Story Structure Properties to Recall of Television Stories in Young Children with Attention-Deficit Hyperactivity Disorder and Nonreferred Peers. *Journal of Abnormal Child Psychology*, *27*(4), 293–309. http://link.springer.com/10.1023/A:1022658625678

Mao, H. H., Majumder, B. P., McAuley, J., & Cottrell, G. W. (2020). Improving Neural Story Generation by Targeted Common Sense Grounding. *arXiv:1908.09451 [cs, stat]*. ArXiv: 1908.09451. http://arxiv.org/abs/1908.09451

McGill, A. L. (2002). Alignable and nonalignable differences in causal explanations. *Memory & Cognition*, *30*(3), 456–468. http://link.springer.com/10.3758/BF03194946

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*. ArXiv: 1301.3781. http://arxiv.org/abs/1301.3781

Niu, F., Zhang, C., Ré, C., & Shavlik, J. (2012). Elementary: Large-Scale Knowledge-Base Construction via Machine Learning and Statistical Inference. *International Journal on Semantic Web & Information Systems*, *8*(3), 42–73. https://doi.org/10.4018/jswis.2012070103

Perales, J. C., Catena, A., & Maldonado, A. (2004). Inferring non-observed correlations from causal scenarios: The role of causal knowledge. *Learning and Motivation*, *35*(2), 115–135. https://linkinghub.elsevier.com/retrieve/pii/S0023969003000420

Pham, H., Dai, Z., Xie, Q., Luong, M.-T., & Le, Q. V. (2021). Meta Pseudo Labels. *arXiv:2003.10580 [cs, stat]*. ArXiv: 2003.10580 version: 4. http://arxiv.org/abs/2003.10580

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, (p. 24).

Richardson, M., & Domingos, P. (2003). Building large knowledge bases by mass collaboration. In *Proceedings of the 2nd international conference on Knowledge capture*, K-CAP '03, (pp. 129–137). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/945645.945665

Schick, T., & Schütze, H. (2020). Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. *arXiv:2001.07676 [cs]*. ArXiv: 2001.07676. http://arxiv.org/abs/2001.07676

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W. R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., & Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, *577*(7792), 706–710. Number: 7792 Publisher: Nature Publishing Group. https://www.nature.com/articles/s41586-019-1923-7

Starr, W. (2021). Counterfactuals. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2021 ed. https://plato.stanford.edu/archives/spr2021/entries/counterfactuals/

Trabasso, T., & van den Broek, P. (1985). Causal thinking and the representation of narrative events. *Journal of Memory and Language*, *24*(5), 612–630. https://linkinghub.elsevier.com/retrieve/pii/0749596X8590049X

Weikum, G., & Theobald, M. (2010). From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, (pp. 65–76).