DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU

HARVARD LIBRARY
Office for Scholarly Communication

Predicting Non-Restrictive Noun Phrase Modifications Through Deep Semantic Analysis

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

Citation	Bal, Mustafa. 2019. Predicting Non-Restrictive Noun Phrase Modifications Through Deep Semantic Analysis. Bachelor's thesis, Harvard College.
Citable link	https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37364649
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

Predicting Non-Restrictive Noun Phrase Modifications Through Deep Semantic Analysis

A THESIS PRESENTED
BY
MUSTAFA BAL
TO
THE DEPARTMENTS OF COMPUTER SCIENCE AND LINGUISTICS

IN PARTIAL FULFILLMENT OF THE HONORS REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS
IN THE SUBJECTS OF
COMPUTER SCIENCE AND LINGUISTICS

HARVARD COLLEGE
CAMBRIDGE, MASSACHUSETTS
MAY 2019

© *2019 - Mustafa Bal*
All rights reserved.

Thesis advisor: Stuart Shieber Mustafa Bal

Predicting Non-Restrictive Noun Phrase Modifications Through Deep Semantic Analysis

Abstract

The difference between restrictive and non-restrictive modifier clauses has been well documented in linguistics. A model that can distinguish non-restrictive modifiers from restrictive modifiers can provide shorter sentences for Natural Language Processing applications3 and improve personal voice assistants in sounding more natural to users. Previous research has provided an annotated corpus and a relatively successful model that predicts non-restrictive modifiers in given sentences. However, this model suffers when faced with prepositional and adjectival modifiers. By utilizing this annotated corpus and learning from previously existing literature, we have made a model that can successfully predict non-restrictive noun phrase modifications through deeper semantic analysis while also performing better with prepositional and adjectival modifiers.

Contents

List of Figures

Acknowledgments

This thesis would not have been possible without the support of several people, whom I wish to acknowledge.

First, I want to thank my thesis advisor Professor Stuart Shieber, who has guided me through the multiple stages of producing a solid piece of research. His patient guidance and expertise have saved me from a myriad of pitfalls in writing, and his help in pointing me toward the right resources and giving me suggestions when I was stuck at multiple uncertain points proved particularly useful.

I would like to express my deepest appreciation to Professor Stephen Chong for his support as a teacher, mentor and friend. It is because of his tireless support and warm personality both inside and outside the classroom that I chose to delve into further Computer Science research. I owe him tremendous gratitude for contributing to my growth as an undergraduate researcher here at Harvard.

I also would like to thank Dr. Diti Bhandra for introducing me to restrictive and non-restrictive noun phrase modifiers and for her guidance in preparing my research in this topic.

Lastly, I express my warmest gratitude to my father Kadir, my mother Nur Ayferi, and my sister Ayşe Rana, for theır unconditional lifelong love, support, and encouragement in all aspects of my studies, as well as to my friends who supported me during the process of writing this thesis.

1 Introduction

1.1 Overview

Linguistic literature through Huddleston et al. [13] defines two different types of modifiers of noun phrases: restrictive modifier clauses and non-restrictive modifier clauses. Consider the following two sentences.

(1.1) He wore the shirt that his father bought him.

(1.2) The mother talked about Bill Gates who just appeared on television.

The underlined phrase in 1.1 is an example of a restrictive modifier clause. Restrictive modifier clauses play a major role in defining the meaning of noun phrases they are attached to, and provide required information about the noun of the sentence. The underlined phrase in 1.2 is an example of a non-restrictive modifier clause. Non-restrictive modifier clauses contribute additional and optional information to an already evident entity.

The difference noted here between restrictive and non-restrictive modifier clauses differ in semantics, and their different effects are evident when the given phrases are removed from the sentences. For example, once we remove the underlined phrase in 1.1 from the sentence, the reader no longer knows that the given male has worn a specific shirt. That extra bit of information that restricted the meaning of the noun (the shirt) is lost. However, when we remove the underlined phrase in 1.2 from the sentence, the meaning of the sentence does not change. The reader still knows who Bill Gates is and does not need the extra bit of information that he recently appeared on television to ascertain which Bill Gates the mother is talking about, or in linguistics-speak, to restrict the meaning of Bill Gates.

1.2 Motivation

With this understanding from Subsection 1.1, it is evident that the removal of non-restrictive noun phrase modifiers from sentences will not affect the meaning of these sentences, and will yield shorter sentences with the exact same semantic meanings. This finding presents itself to be very useful when applied to computational tasks that deal with text. For one, automatically removing non-restrictive noun phrase modifications while preserving restrictive noun phrase modifications in sentences result in sentences with shorter noun-phrases, which is advantageous in compressing sentences in digital files [14]. This exact approach is also useful in obtaining summaries of large bodies of text to yield short summaries, which is also known as abstractive summarization [9].

In addition, in semantic-role-labelling tasks that consist of identifying the semantic roles of words in a given sentence, research has shown that sentences with more words result in semantic-role predictions that are less accurate than those in shorter sentences. By identifying and removing non-restrictive noun phrase modifications from longer sentences, the prediction accuracy rates of these semantic models can be increased [3]. These three given usage cases belong to the sub-field of Natural Language Processing (NLP), and the identification of non-restrictive noun phrase modifications in both usages result in removing data that would not significantly help an NLP machine-learning model to learn, thereby decreasing the training times required for these models. As such, a successful predictive model that can distinguish non-restrictive noun phrase modifiers from restrictive modifiers can help many NLP applications that are related to text analysis.

In addition, it has been noted by recent research that with sentences that are answers to posed questions, if these answers that do not contain non-restrictive noun phrase modifications, then they are considered more concise answers [26]. This can be interpreted in that sentences that do not contain non-restrictive noun phrase modifications sound more natural to the human ear. This finding can play a role in helping personal voice assistants, such as Apple's Siri, Google's Assistant [4], and Amazon's Alexa platforms, sound more like human speech rather than speech that has obviously been generated by a machine.

When a query is given to such personal assistants, humans usually attach non-restrictive noun phrase modifiers to the subject or the object of the sentence. When a personal voice assistant repeats these non-restrictive modifiers, it does not repeat it because it narrows the subject or object in question, but because the user phrased his or her question that way. In this case, the personal voice assistant does not need to repeat the non-restrictive noun phrase modifiers, as the user already knows the subject or object in question, and does not need any modifier to provide extra information on the subject or object in question.

Up until today, there have been few attempts made in annotation the restrictiveness of noun-phrases, which all varied in terms of complexity and success of predicting non-restrictive noun phrase modifiers. The first of these attempts was a simple rule check, which was that a noun phrase modifier is predicted to be non-restrictive if and only if this modifier is preceded by a comma [12]. This simple rule was chosen by these authors as they noted that in the English literature, noun phrase modifiers that are non-restrictive tend to be preceded by commas. Examples of this trend can be found in Section 2.1. This simple rule proved to be successful as it was the best non-restrictive noun phrase modification predictor to date, even though there was no usage of any classifier machine learning models used.

The comma rule mentioned above held its ground until the second biggest study to study the prediction of non-restrictive noun-phrase modifiers came out. The second study utilized a machine learning model that utilized the Conditional Random Field class of statistical modeling to predict non-restrictive noun phrase modifiers given in sentences. This model was trained on a small corpus they created, which consisted of sentences that indicated the location and restrictiveness of marked noun phrase modifiers. This new contribution to predicting the restrictive state of noun phrase modifiers reigned supreme until the latest best study on this subfield was published.

The biggest and most recent body of work has been done by Dr. Gabriel Stanovsky, advised by Prof. Ido Dagan at Bar-Ilan University. Stanovsky and Dagan devised a crowd-sourcing annotation methodology, and brought together a large scale corpus, with which they trained their CRFsuite model that identifies non-restrictive modifiers with notable improvements over prior methods. They have made their annotated corpus available to aid further research done in this field, and more importantly, have built a model that identifies non-restrictive noun phrase modifications with notable success. However, as Stanovsky and Dagan acknowledge, their model does poorly with certain types of modifier phrases, such as prepositional and adjectival modifiers. In the "Conclusions and Future Work" section of their paper, they note that further semantic analysis is required to "develop classifiers that deal better with prepositional and adjectival modifiers" [26].

1.3 Contribution

The main objective of this thesis is to demonstrate that deeper semantic analysis is required to better predict prepositional and adjectival noun phrase modifiers, as well as other types of modifiers such as relative, non-finite, verbal and appositive, than currently exists in present-day literature. We contribute into the problem of predicting non-restrictive noun phrase modifications in three ways:

1. We compile the current-day literature on predicting the restrictiveness of noun phrase modifiers. In doing this, we provide an overview of noun phrase modifiers, detail the varieties of restrictive and non-restrictive noun phrase modifiers, metrics for evaluating predictive models of noun phrase modifiers, and the main idea behind classifier machine learning models.

2. We reproduce the results of previous works that attempted to distinguish non-restrictive noun phrase modifiers from its restrictive counterparts, and confirm the improvement in metrics that occur between each successive and newer studies.

3. We introduce our own machine learning model that utilizes deeper semantic analyses to achieve higher scores in the accuracy, precision, recall and F1 benchmarks, particularly with prepositional and adjectival noun phrase modifiers, but also with other types of modifiers such as relative, non-finite, verbal and appositive.

This thesis will be structured as follows. Chapter 2 introduces the relevant background behind noun phrase modifiers and the metrics that will be useful in determining the success of the non-restrictive noun phrase modification classifier machine learning models, in addition to explaining the main classifier machine learning model used throughout this thesis. Chapter 3 goes into detail with related works of machine learning models that can distinguish the restrictiveness of noun phrase modifiers, where the works range in complexity from simple character equivalence checks to utilizing complex statistical machine learning models. Chapter 4 explains the implementation of the machine learning models of both the existing literary research and of our own for predicting non-restrictive noun phrase modifications. Chapter 5 provides detailed observations on the obtained testing results of the existing research models as well as of our own model. Finally, Chapter 6 delivers closing remarks and conclusions on the thesis. The research done in this thesis utilizes the English language along with its accompanying grammar and semantics.

2
Background

In Chapter 2, we describe the different types of modifier clauses, and related works that make up the background of this thesis. First, we present the six types of noun phrase modifiers that differ in their restrictiveness. We then look into the metrics that will be useful in measuring the success of our classifier machine learning models which will be used for predicting non-restrictive noun phrase modifications from restrictive noun phrase modifications.

2.1 Noun Phrase Modifiers

In this thesis, as stated in Chapter 1, we mirror the approach taken by Stanovsky et al. [26] and follow Huddleston et al.'s distinction [13] between the two different types of noun phrase modifiers. These two types are restrictive modifiers that add inseparable meaning to the greater noun phrase and

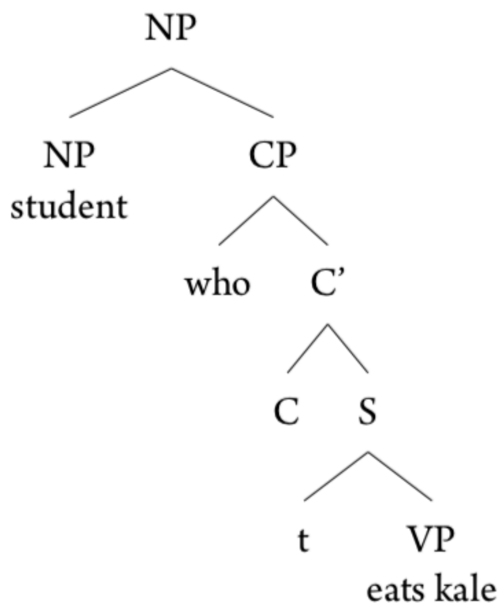

Figure 2.1.1: Sentence diagram of the noun phrase of 2.1.

non-restrictive modifiers that add auxiliary and separate information to the noun phrase. While the two types of noun phrase modifiers might not seem too different at first, their semantic differences play a big role in determining the meaning of nouns they are modifying.

From here on out, we will address restrictive noun phrase modifiers as RMs, and non-restrictive noun phrase modifiers as NRMs.

We will now observe two examples of noun phrase modifiers. Let us first observe this first set of examples [1]:

(2.1) (Sentence) John is a student who eats kale.

(2.2) (Noun Phrase) student who eats kale

(2.3) (Relative Clause) who eats kale

We see that 2.2 is the noun phrase inside 2.1. In this case, this noun phrase is a subject relative clause, where the noun of the sentence is modified by the relative clause. This relative clause is an example of a restrictive relative clause, as it narrows the scope of the student from any possible person who happens to be a student to a specific student who eats kale. Unlike a non-restrictive relative clause, the removal of this relative clause will result in John becoming an ordinary student where we do not possess the knowledge of him as someone who eats kale or not.

Let us observe another set of examples [1]:

(2.4) The book which John recommended is good.

(2.5) The book, which John recommended, is good

(2.6) (Common Relative Clause) which John recommended

A keen reader would note that the only difference between 2.4 and 2.5 is the comma between the words 'recommended' and 'is'. Indeed, that is the only input difference; however, the difference of the meanings that occur as a result of this is significant. The same relative clause 2.6 that appears in both 2.4 and 2.5 have different restrictiveness states; the relative clause in 2.4 is restrictive whereas the same relative clause in 2.5 is non-restrictive.

As a result, the meaning of 2.4 becomes to be that there is exactly one book in the sentence topic, whereas in 2.5 there can exist multiple books in the sentence topic as long as John particularly recommended only one of the books in the discourse. Such is the power of a comma, and consequently, the power of the restrictiveness of noun phrase modifiers.

Let us now express the interpretation of restrictive and non-restrictive relative clauses by utilizing semantic representation of sentences containing restrictive and non-restrictive relative clauses, which Egg provides [6].

(2.7) The train which leaves at 11:30am is waiting on platform 5.

(2.8) The train, which leaves at 11:30am, is waiting on platform 5

(2.9) (Common Relative Clause) which leaves at 11:30 am

The meaning of 2.7 is that there may be more than one train that leaves that train station from platform 5, in addition to the train that leaves at 11:30 am. The common relative clause is necessary in 2.7 to distinguish the train that leaves at 11:30 am from trains that leave at other time points in the day. Compared to this, the meaning of 2.8 is that is only one train that leaves the train station and is waiting on platform 5, that also happens to be leaving at 11:30 am.

Now, we observe the semantic representations of 2.7 and 2.8, respectively.

(2.10) $\exists! x[\text{train}(x) \wedge \text{leave-at-11:30am'}(x)] \wedge\ \text{wait-on-p5'}(x)$

(2.11) $\exists! x[\text{train}(x)] \wedge \text{leave-at-11:30am'}(x) \wedge\ \text{wait-on-p5'}(x)$

In 2.10, we can see that there is a unique element that is classified as a train and that this unique element leaves at 11:30 am. In other words, this means that there is a unique train that leaves at 11:30 am, and that there may be other trains that leave at different times. In comparison, in 2.10, there is only one train that leaves from the train station. It is true that this one train leaves at 11:30 am, however, the information on the time of the train's departure is not required to identify the one unique train.

Thus, by utilizing the semantic representation of the sentences, we can see that in 2.7 ,with 2.10 as its semantic representation, is the sentence with 2.9 being the restrictive noun phrase modifier, and that in 2.7 ,with 2.11 as its semantic representation, is the sentence with 2.8 being the non-restrictive noun phrase modifier.

Certain noun phrase modifiers such as determiners and genitives are always restrictive modifiers, while other syntactic modifiers such as relative, prepositional, adjectival, verbal and non-finite modifiers can either be interpreted as restrictive or non-restrictive modifiers. [13].

We now dive deeper into the six types of noun phrase modifiers with examples (the first four given by Huddleston et al. [13], the latter from Rower [22]) for these modifiers and state their restrictive or non-restrictive states. For each sentence, the noun phrase modifiers are underlined, and justifications for their restrictive or non-restrictive states are provided.

2.1.1 Relative Modifiers

(2.12) (RM) The necklace that her mother gave her is in the safe.

(2.13) (NRM) The governor disagreed with the U.S ambassador to China who seemed nervous.

Here, the noun phrase modifier 2.12 is restrictive, as the modifier reduces the possible set of necklaces that are deemed to be in the safe. The necklace in question has to fit the criteria that the necklace was given by the mother of the daughter present in the context.

For 2.13, the fact that the U.S. ambassador to Chine seemed nervous does not clarify the identity of the ambassador. The underlined modifier is superfluous information in identifying the ambassador. As such, the underlined modifier in question is non-restrictive.

2.1.2 Non-finite Modifiers

(2.14) (RM) People living near the site will have to be evacuated.

(2.15) (NRM) The sheriff Arthur Lester, standing against the wall, looked tired.

In 2.14, the underlined modifier restricts the group of people that will need to be evacuated. These group of people change from any collection of humans who might live all across the world to those that are only living near the site. So, the underlined modifier is a restrictive noun phrase modifier.

In 2.15, the information that the sheriff is standing against the wall does narrow down. The identity of the sheriff is already known as Arthur Lester, and it is told to the reader already that he looked tired. As a result, the underlined modifier is a non-restrictive noun phrase modifier.

2.1.3 Prepositional Modifiers

(2.16) (RM) The kid from New York won the lottery.

(2.17) (NRM) The assassination of Franz Ferdinand from Austria started World War 1.

For 2.16, the identity of the kid who won the lottery changes from any kid living in the world to a kid that is living in New York. While the information given is not enough to restrict the identity of the kid to a specific person, the modifier given does narrow the pool of people eligible to be this kid. Thus, the provided modifier is a restrictive noun phrase modifier.

For 2.17, the information that Franz Ferdinand is from Austria does not add extra info to the identity of the person who was assassinated; this person is already given as Franz Ferdinand. Since his nationality does not provide extra information in this case, the provided modifier is a non-restrictive noun phrase modifier.

2.1.4 Adjectival Modifiers

(2.18) (RM) The good boys won.

(2.19) (NRM) The water level rose a good twelve inches.

With 2.18, the list of boys is restricted from any grouping of males to those who are considered "good". In this context, the exact semantic meaning of "good" is enough to restrict those who are not considered "good". As a result, the underlined modifier is a restrictive noun phrase modifier.

Considering 2.19, the adjective "good" does not contribute to the exact definition of the measurement of twelve inches, as twelve inches is always equal to twelve inches. As such, this modifier is a non-restrictive noun phrase modifier.

2.1.5 Verbal Modifiers

(2.20) (RM) On Friday the Pit Committee discussed the rules concerning off-duty hissing.

(2.21) (NRM) The Pit Committee's new off-duty hissing guidelines, setting out more precise standards for choral harmony, were received with jubilant writhing.

For 2.20, the concern for off-duty hissing acts in identifying the rule discussed by the Pit Committee. The set of rules the Pit Committee could have been discussing about is made smaller, as the possible contents of this set go from all possible rules there may exist to those that only concert the action of hissing while off-duty. As a result, the underlined modifier is a restrictive noun phrase modification.

With 2.21, the underlined modifier does not restrict the context of the aforementioned guidelines. In fact, the phrase "new off-duty hissing" identifies the guidelines put forward by the Pit Committee. The underlined phrase merely provides more information about the guidelines. As a result, the underlined modifier is a non-restrictive noun phrase modification.

2.1.6 Appositive Modifiers

(2.22) (RM) Fang was amused by the fact that writhing breeches are no longer fashionable.

(2.23) (NRM) Sidonia, the well-known hieroglyphic dancer, will be performing during the migration.

For 2.22, the underlined modifier serves to identify the exact fact that humored Fang. As a result, the list of facts that could have amused Fang is narrowed. Because of this, the underlined modifier is a restrictive noun phrase modifier.

With 2.23, the underlined modifier provides more information about the dancer. However, the identity of the dancer is already known, who is Sidonia. As the information given does not serve to restrict the set of possible candidates that could be the well-known hieroglyphic dancer, the underlined modifier is a non-restrictive noun phrase modifier.

We can see that all six of these modifier types can have restrictive and non-restrictive properties. These same six modifier types are the same types that we will later see in the Implementation section of our machine-learning model as well as previous models from the Literature Review section for distinguishing restrictiveness of noun phrases in sentences. However, to first measure the success of these models, we need to be aware of the metrics we will be using.

2.2 Metrics for Evaluating the Success of Predictor Models

It is important to be able to compare the success of different machine learning models that are aimed at completing the same task. A more successful machine learning model will have higher metric numbers than lesser successful models. Different metrics can point to various strengths and weaknesses of such machine learning models.

The task of a machine learning algorithm that is attempting to predict the restrictiveness of a noun phrase modifications is a binary classification problem. In a binary classifier problem, there are only two possible state choices that exist. These two states are ”positive” and ”negative”.

In our case, the **positive** state will indicate that a given noun phrase modification is a non-restrictive noun phrase modification, and a **negative** state will indicate that the given noun phrase modification is restrictive noun phrase modification. We now can compare the predicted and actual states of the given noun-phrase modifications by introducing **true** and **false** states. A state is considered true if its predicted and actual states match, otherwise it is considered false.

We now can define our four possible combined states, where again, NRM stands for non-restrictive noun phrase modifier and RM stands for restrictive noun phrase modifier:

- True Positive (tp): Predicted NP modifier is NRM, actual NP modifier is RM
- False Positive (fp): Predicted NP modifier is NRM, actual NP modifier is RM
- True Negative (tn): Predicted NP modifier is RM, actual NP modifier is RM
- False Negative (fn): Predicted NP modifier is RM, actual NP modifier is NRM

As we now have defined our four composite states, we can make sense of metrics that exist to provide meaningful comparative data on our machine learning models. There are four metrics that exist for evaluating the success of a classifier machine learning model: accuracy, precision, recall, and F1. We will be utilizing the standards that Powers [21] utilizes. We will now explore these four metrics.

2.2.1 Accuracy

The most simple metric that exists for evaluating the success of a machine learning classifier model is the Accuracy metric. This metric focuses on the correct positive and negative predictions made by the machine learning model. Expressed formally:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

From the equation, we can see that the Accuracy metric is as its highest when there are a lot of true positives and true negative results. While it may be good to have a large amount of true negatives, if the quantity of true negatives is significantly higher than the quantity of true positives, then this will cloud the weaknesses of our classifier machine learning model.

The Accuracy metric does have its use cases, in that it can provide concise info on how many true predictions have been made by the machine learning model. However, judging a classifier machine learning model just by the accuracy metric is not only disadvantageous, but also very risky. For one, the Accuracy metric does not function well with data sets that are imbalanced. Such an imbalanced data set may be when there are an overwhelming amount of positive samples, and a very few amount of negative samples. In this situation, a high accuracy metric can easily be achieved when all samples are predicted to be positive.

As such, there is a need to utilize the Accuracy metric with caution. We can take precaution with the Accuracy metric in many ways, and to avoid the problematic predictions the Accuracy metric may provide, we utilize other metrics that will provide more meaningful information about the success of our classifier machine learning models. We will now learn about the Precision metric.

2.2.2 Precision

The Precision metric provides information on how precise a classifier machine learning model is. In other words, the Precision metric shows how many of the chosen items from a given list are relevant to the current context. The equation of the Precision metric is as follows:

$$\text{Precision} = \frac{tp}{tp + fp}$$

In a nutshell, the Precision metric calculates the ratio of the total number of true positive results over the number of total predicted positive results. The importance of the Precision metric is high when our classifier machine learning model is heavily punished for having lots of false positive results.

The main strength of the Precision metric is that it can show how large the quantity of False Positives are. In our context where we are trying to determine non-restrictive noun phrase modifications, the Precision metric will show if we incorrectly predict too many restrictive noun phrase modifications as actually non-restrictive. Thus, we will know in our context that a classifier machine learning model that has a higher Precision metric is one that does not incorrectly predict multiple restrictive noun phrase modifications as non-restrictive noun phrase modifications.

We now will look at our third type of metric, which is the Recall metric.

2.2.3 Recall

Similar to but not quite the same as the Precision metric, the Recall metric provides information on how many of the selected items from a list are actually relevant to the current context. The equation of the Recall metric is as follows:

$$\text{Recall} = \frac{tp}{tp + fn}$$

In informal words, the Recall metric calculates the total number of true positive findings over the total amount of actual positive findings, which is equal to the sum of the quantity of true positive and false negative results. With this understanding, the Recall metric will be especially useful if we are trying to determine the best classifier machine learning model that results in the lease number of False Negatives, that is, the case where the prediction of the model is that the noun phrase modification is restrictive when it is actually non-restrictive.

We now progress onto our fourth type of metric, which is the F1 metric.

2.2.4 F1

The F1 metric is a function of the two previous metrics, Precision and Recall. The formula for the F1 metric is as follows:

$$F_1 = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The purpose of the F1 metric is to utilize a way of measurement of the success of a classifier machine learning model that finds an optimal balance between the

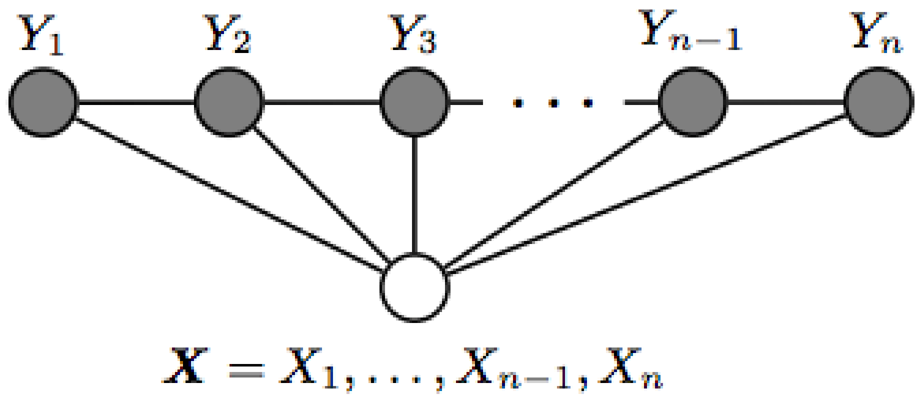

Figure 2.3.1: Graphic representation of a chain-structured Conditional Random Field for sequences [27]

Precision and Recall metrics. The optimal balance between the Precision and Recall metrics isn't simply theorized, but is actually equivalent to the harmonic average of the Precision and Recall metrics. A machine learning model with a high F1 score indicates good Precision and Recall metrics, whereas the opposite would signal poor Precision and Recall metrics.

Now that we have reviewed the metrics for evaluating classifier machine learning models, we will now review the main statistical modeling method we will use in this thesis: Conditional Random Fields.

2.3 Conditional Random Fields for Classifier Models

There are two major categories of machine learning models: discriminative and generative. For our purposes of classifying two types of noun phrase modifiers, restrictive and non-restrictive, it is in our interest to utilize discriminative machine learning models that are able to clarify the boundary between two types of classes. This way, such a model can succeed at predicting the most likely

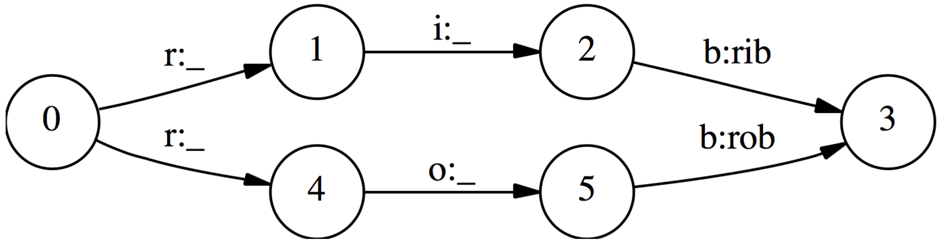

Figure 2.3.2: The Label Bias problem, as taken from Lafferty et al. With the given observation sequence $r\ i\ b$, states with one outgoing transition are ignoring their observations. [15]

sequence of labels from a given sequence of inputs. Currently, one of the most successful and reliable models in modeling sequential data are Conditional Random Fields (CRF), as shown in 2.3.1. [15].

Formally, Lafferty et al. define a Conditional Random Field as follows, with certain preconditions:

- $\mathbf{X}$ is a random variable over data sequences that will be labeled.
- $\mathbf{Y}$ is a random variable over corresponding label sequences
- Random variables $\mathbf{X}$ and $\mathbf{Y}$ are jointly distributed. But in a discriminative framework, we have the conditional model $\mathbf{p}(\mathbf{Y}|\mathbf{X})$ from paired observation and label sequences.

Definition (Conditional Random Field)**.** .

Let $G = (V, E)$ *be a graph such that* $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$*, so that* $\mathbf{Y}$ *is indexed by the vertices of G. Then,* $(\mathbf{X}, \mathbf{Y})$ *is a conditional random field when the random variables* $\mathbf{Y}_v$*, conditioned on* $\mathbf{X}$*, obey the Markov property with respect to the graph:*

$p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \leftrightarrow v)$*, where* $w \leftrightarrow v$ *means that* w *and* v *are neighbors in G.*

The strength of Conditional Random Fields lie under two fundamental principles. First, CRFs do not depend heavily on the assumption of independence. In a lot of machine learning models, it is assumed that variables do not depend on each others' values, and that their values do not affect each other in any tangential way. This assumption is not always useful, and sometimes in certain cases such as those faced in the field of computational semantics, such an assumption may lead to large inaccuracies. Second, CRFs do not suffer from the Label Bias problem as shown in 2.3.2. This bias occurs when the probability of a future state is only depending on a given observation and the value of the previous state. CRFs solve this by ignoring local per-state normalizations that occur due to immediately previous states and instead using global per-sequence normalizations [15]. This is due to the fact that CRFs are undirected graphs, and most models that suffer from the Label Bias problem are directed graphs. .

There exists multiple varieties of Conditional Random Fields, such as Semi Markov Conditional Random Fields [23] that assign labels to sub-sequences of input segments instead of assigning them to individual elements. While such varieties do exist and offer slightly better performance differences, for the purpose and context of this thesis in predicting non-restrictive noun phrase modifiers, we will stick to using traditional Conditional Random Fields and introduce features that build off of CRFs to better focus on semantic analysis of noun phrase modifiers than the underlying statistical models that predicts noun phrase modifiers.

As a result of CRF's great potential in modeling sequential data, they are widely used in numerous Natural Language Processing tasks. One area of application CRFs are very successful in is Shallow Parsing . As Sha and Peraira [25] demonstrate, Conditional Random Field models perform significantly better than generative models like Hidden Markov Models at performing Shallow Parsing of sentences at identifying constituents that make up a sentence, such as nouns, adjectives, verbs, etc..., and connects these to higher order grammar trees. Another area CRFs are strong in is Named Entity Recognition, where Sato et al. [24] demonstrates how linear chain CRFs accurately mark textual named entities into pre-defined categories such as names of locations, organizations, persons, as well as expressions of percentages and quantities. While these applications of CRFs already exist, we will be using them to predict non-restrictive noun phrase modifications.

We now progress into past works that have attempted to mark the restrictive states of these modifiers, starting from simple determiners of non-restrictive noun phrase modifiers and moving towards trained machine learning models that utilize Conditional Random Fields.

3

Literature Review

In this chapter, as previously mentioned, there exist only a few research pieces that delved into marking the restrictiveness of noun phrase modifiers. We will now explore the progress made in ascending order by date of publication.

3.1 The Comma Rule

The first research publication that directly addressed the restrictiveness of noun phrase modifiers is Honnibal et al.'s "Rebanking ccgbank for improved np interpretation" paper [12]. Their rule for determining the restrictiveness of a noun phrase is very simple: a given modifier is considered non-restrictive if and only if that modifier is preceded by a comma. They tested their theory on a corpus they compiled from the CCGBank corpus [11], where they added numerous automated restrictiveness annotations to noun phrase modifiers in the corpus to provide input for their simple classifier model.

It is not hard to see the success if this simple yet effective rule with certain sentences. For example, the modifiers in 2.4 and 2.5 can easily be distinguished by the comma rule. Similarly, Honnibal et al. found that their rule works relatively well with relative noun phrase modifications. However, this rule does not work with most of the other types of modifiers, as they do not need to be preceded by commas to be considered restrictive. Examples 2.12 through 2.23 are great counter-examples to the comma rule in this regard. However, Honnibal et al. did successfully call attention to the need of finding an accurate way of predicting the restrictiveness states of noun phrase modifications, and further publications better addressed this challenge.

3.2 Dornescu et al.'s Simple Predictive Model

The first paper that utilized a machine learning model specifically to distinguish between restrictive and non-restrictive noun phrase modifiers is that of Dornescu et al.'s "Relative clause extraction for syntactic simplification" [5]. This paper was a major breakthrough in that it made use of machine learning models to predict non-restrictive modifiers rather than utilizing simple if-else conditions.

To train their CRF model, Dornescu et al. compiled a corpus of sentences that include noun phrase modifiers. This corpus was compiled manually by trained annotators who did the following three actions per sentence:

1. Specified the beginning and ending locations of noun phrase modifiers.
2. Indicated the type of noun phrase modifiers (as explained in Section 2.1
3. Marked the included noun phrase modifiers as restrictive or non-restrictive.

After compiling this corpus, Dornescu et al. utilized multiple supervised and unsupervised machine learning methods to predict the restrictiveness of given noun phrase modifiers. Their best performing model was the supervised

Conditional Random Field approach (CRF) through an open source CRF classifier named CRFSuite [19] that utilized features common in semantic chunking applications, such as the part-of-speech tags, word form and lemma of given inputs. While their best performing model was more successful than that of Honnibal et al., there is yet one more piece of research that beat Dornescu et al.'s method.

3.3 Stanovsky et al.'s Extended Predictive Model

The non-restrictive noun phrase modifier predictive model of Stanovsky et al. is the latest literature offers in terms of annotating noun phrase modifications. There are two main contributions of Stanovsky et al.'s research: (1) a robust machine learning model that can identify non-restrictive modifiers with significant improvements over previous models, and (2) a large scale corpus for training the aforementioned machine learning model and a novel crowd-sourcing corpus annotation method [5]. We will now dive deeper into the machine learning model of Stanovsky et al., and explore their annotated corpus in Section 2.3.

The machine learning model of Stanovsky et al. builds on the same basics that Dornescu et al.'s model is based on: the CRF classifier CRFsuite. It also obtains its first two features from previous literature: the preceding comma rule from Honnibal et al., and the chunking-application features from Dornescu et al. In addition to extending these two features in their own machine learning model, Stanovsky et al. add in their own features:

1. Enclosing commas: Gaining inspiration from common writing styles and grammar rules, the noun phrase modifier is more likely to be non-restrictive if the modifier is both preceded and concluded with commas.

2. Named entities: Stanovsky et al. argue that the modifiers of named entities are more likely to be non-restrictive. By incorporating Finkel et al.'s

Named Entity Recognizer (NER) [7], the modifier can be checked to see if NER recognizes the named entity, and mark the modifier as more likely being non-restrictive.

3. Lexical word embedding: Mikolov et al. [18] has built a model that provides pre-trained word embeddings where the input is the head word of the chosen modifier. This way, the model can more easily distinguish the restrictive state of similar noun phrase modifiers such as ”good” and ”fine”.

4. Modifier type: Stanovsky et al. also added the type of the modifier that is being analyzed as a feature of their classifier model to link certain features as indicative for given types of modifiers. For example, certain adjectival modifiers have specific word embeddings that can identify the modifier's restrictiveness.

The combination of these features enable the machine learning predictor of non-restrictive noun phrase modifications of Stanovsky et al. perform the best out of the three reviewed papers.

3.4 Annotated Corpus by Modifier Restrictiveness

Stanovsky et al. also provide a large scale crowdsourced annotated corpus that includes ”linguistically motivated features which are robust enough to perform well over automatically predicted parse trees” [26]. This corpus includes 744 sentences for training, 249 for development purposes, and 248 sentences for testing a prediction model.

The sentences are from the 2009 Computational Natural Language Learning Conference English data set [10]. These sentences are from news articles that are annotated by the projects of PropBank [20], NomBank [17], and TreeBank [16]. While the sentence text files are given by the aforementioned sources, these text files do not include information on given restrictive and non-restrictive noun phrase modifiers of the sentences.

Before we go into further detail on the strengths of the highly accurate annotated corpus, we must review the important metric that will tell us the degree of agreement between two sets of data. This measure is known as Cohen's kappa [8]. The equation of Cohen's kappa is below.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

$$p_o = \text{relative observed agreement between raters}$$

$$p_e = \text{hypothetical probability of agreement by chance}$$

Depending on the value of Cohen's kappa, the agreement levels vary as shown below.

- κ = 0-0.1: agreement due to chance
- κ = 0.1-0.20: slight agreement
- κ = 0.21-0.40: fair agreement
- κ = 0.41-0.60: moderate agreement
- κ = 0.61-0.80: substantial agreement
- κ = 0.81-0.99: near perfect agreement
- κ = 1: perfect agreement

Now, we return to the the limitation of acquiring an annotated corpus of restrictive and non-restrictive noun phrase modifiers. Stanovsky et al. devised a process to annotate this large corpus of sentences based on its noun phrase modifiers' restrictiveness. Their objective was to reliable label the restrictive states of the known noun phrase modifiers as best as possible, to the point where this labelling at a massive scale should match or at least be very close to the labelling accuracy that a pair of expert human annotators would provide.

Stanovsky et al. approached this problem by first obtaining a baseline to compare their to-be crowdsourced annotated corpus. This baseline consists of 100 sentences that contain 2019 modifier of noun phrases, where the restrictiveness of the noun phrase modifiers were annotated by two researchers who are proficient in natural language processing and linguistics. For every noun phrase in the expert annotation corpus, the researchers were requested to mark the modifier as either restrictive or non-restrictive according to the linguistic definition of restrictive and non-restrictive noun phrase modifications given by Huddleston et al. [13]

To get the most correct expert annotations possible, the authors made the two researchers label the restrictiveness of the given noun phrase modifiers independently and then checked the agreement of the expert classifications. The authors utilized two sets of metrics for measuring the success of the two researchers: the percentage of matches the two researchers got the same, and Cohen's kappa that measures for non-accidental agreement between outputs. Stanovsky et al. found that the annotation agreement was 93.5 %, and with a κ score of 84.2 %. Both of these measures indicate that the annotation by the researchers have been very successful. Stanovsky et al. note that the few disagreement the two researchers did have were caused by semantic ambiguities, which happened when two correct readings of sentences resulted in different annotations. For example, take the sentence below:

(3.1) Sympathetic fans have sent Ms. Shere copies of her recipes <u>clipped from magazines over the years</u>.

For this specific example, one researcher read the underlined modifier as a restrictive noun phrase modifier, as they thought that this was identifying particular recipes from another larger collection of recipes. The other researcher read the underlined modifier as a non-restrictive noun phrase modifier, as they thought the underlined part only providing extra information on the recipes that have been sent.

After compiling the expert-annotated corpus, Stanovsky et al. then devised a crowdsourcing annotation procedure to label the restrictiveness of noun phrase modifications by utilizing the brainpower of everyday people. This procedure involved assigning a binary label,**true** for non-restrictive and **false** for restrictive, to a 4-tuple (s, v, p, m) where s is the sentence, v is the verbal predicate containing a noun phrase p, and m is the modifier of the noun phrase. Then, for every tuple, the reader is presented a question with either true or false answers. This question is provided from the verbal predicate v, which can be obtained for a simple argument role question, and inquires about the noun phrase p in the sentence s, and asks if the noun phrase p without its modifier m can answer the question in the same way the noun phrase p with its modifier m can. Consider the following example: By asking a binary question that is trying to get the annotator to see if two given answers are semantically the same answers, the annotators of the crowdsourced corpus did not have to be people who are trained in the formal linguistic definition of restrictive and non-restrictive noun phrase modifications. To demonstrate this, take the following example.

(3.2)

- s = "The speaker thanked President Obama who just entered the room." (Sentence)
- v = "thanked President Obama who just entered the room" (Verbal predicate)
- p = "President Obama who just entered the room" (Noun phrase)
- m = "who just entered the room" (Modifier of noun phrase)
- Question asked to annotator: "Whom did someone thank?"
  - Correct answer: p, "President Obama who just entered the room"
  - Does p - m, "President Obama", also give the correct answer?
    - If **true**, m is a <u>non-restrictive</u> noun phrase modifier.

    * If **false**, *m* is a restrictive noun phrase modifier.
  - In this case, *p* - *m* also is the correct answer.
  - Thus, the modifier *m* "who just entered the room" is a non-restrictive noun phrase modifier.

In example 3.2, the given noun phrase modifier is a non-restrictive modifier as the proposed question can be answered with the noun phrase that does not contain the modifier. Let us take another example.

(3.3)

- *s* = "She wore the necklace that her mother gave her." (Sentence)
- *v* = "wore the necklace that her mother gave her" (Verbal predicate)
- *p* = "the necklace that her mother gave her" (Noun phrase)
- *m* = "that her mother gave her" (Modifier of noun phrase)
- Question asked to annotator: "What did someone wear?"
  - Correct answer: *p*, "the necklace that her mother gave her"
  - Does *p* - *m*, "the necklace", also give the correct answer?
    * If **true**, *m* is a non-restrictive noun phrase modifier.
    * If **false**, *m* is a restrictive noun phrase modifier.
  - In this case, *p* - *m* also is not the correct answer, as the specific identity of the necklace in question is now not certain.
  - Thus, the modifier *m* "that her mother gave her" is a restrictive noun phrase modifier.

Utilizing this crowdsource annotation method, Stanovsky et al. utilized Amazon Mechanical Turk [2] to annotate a total of 2191 modifiers for their restrictiveness. On the Amazon Mechanical Turk platform, people who have access to a computer and an internet connection can complete short tasks for small monetary payments. This process yielded a Cohen's kappa score of $\kappa = 73.9$, which indicates a substantial agreement finding. Similar to the expert

Modifier Type	Amount in Corpus	Ratio of NRMs	Agreement %	Agreement κ
Adjectival	684	41.36%	87.36%	74.70
Prepositional	693	36.22%	85.10%	61.65
Appositive	342	73.68%	80.00%	60.29
Non-Finite	279	68.82%	86.48%	71.04
Verbal	150	69.33%	100.00%	100
Relative	43	79.07%	100.00%	100
Total	**2191**	**51.12%**	**87.00%**	**73.79%**

Figure 3.4.1: Statistics of the crowdsource annotated corpus. The agreement levels, with the matching ratio and Cohen's κ, is displayed for the expert-annotated data between the expert and crowdsource annotations.

annotation case, the modifier types that had the lowest agreement levels were propositional and appositive modifiers with 61.65 % and 60.29 %, respectively. The full statistics of the crowdsource annotated corpus can be found below.

We will be utilizing this crowdsource-compiled corpus for training and testing the performance of the models previous discussed, as well as training and testing our own machine learning non-restrictive noun phrase modifier predictor model. We now move into the implementation details for our own model.

4

Deeper Semantic Analysis for Restrictiveness

In Chapter 4, we detail the features we have added to our own classifier machine learning model for predicting non-restrictive noun phrase modifiers. These additional features are intended to produce a model that is more successful, in terms of metrics described in Chapter 2, than the best model from literature, which is that of Stanovsky et al. As the contributions made by Stanovsky et al. are very significant, in addition to our novel features, it is in our best interest to include all of the features present in the model of Stanovsky et al. into our model as well. This way, we can utilize the strengths of past successful models while adding new dimensions of thought to that of our own.

As Stanovsky et al. note in their paper, their model does not perform well with annotating the restrictiveness of adjectival and prepositional modifiers. The following sections expand on our motivations for adding these specific features to our model.

4.1 N-GRAMS

The first feature we have added to our novel classifier machine learning model is a collection of n-grams features. In the simplest terms, a sequence of words that is of length *n* is known as an n-gram. For example, ”a toy” is a 2-gram, ”a big toy” is a 3-gram, and a ”a big red toy” is a 4-gram.

In our model, we have integrated the n-gram logic to include bigrams (2-grams) and trigrams (3-grams) of the features described in 4.1.

(4.1) **N-gram Feature Algorithm**

- For word at position x in a sentence of length N:
  - if $x > 1$ and $x < N$
    - (-1_word.lower, 0_word.lower, +1_word.lower)
    - (-1_POStag, 0_POStag, +1_POStag)
    - (-1_lemma, 0_lemma, +1_lemma)
    - (-1_1modifierType, 0_1modifierType, +1_1modifierType)
  - else if $x > 1$ (first word in sentence)
    - (0_word.lower, +1_word.lower)
    - (0_POStag, +1_POStag)
    - (0_lemma, +1_lemma)
    - (0_1modifierType, +1_1modifierType)
  - else (last word in sentence)
    - (-1_word.lower, 0_word.lower)
    - (-1_POStag, 0_POStag)
    - (-1_lemma, 0_lemma)
    - (-1_1modifierType, 0_1modifierType)

4.2 Position of Prepositional and Adjectival Noun Phrase Modifiers

The second feature we have added to our novel classifier machine learning model is a distance factor of prepositional and adjectival noun phrase modifiers from the beginning of the sentence. We hypothesize that the farther away the given prepositional or adjectival noun phrase modifier is from the beginning of the sentence, there is a higher chance that that modifier is a non-restrictive noun phrase modifier.

In deciding the relative position of our noun phrase modifiers, we divide our sentence into four quadrants of equal size. The total size of the sentence is equal to the number of words given in that sentence, the quadrant closest to the beginning of the sentence is quadrant 1, and consequently the quadrant farthest from the beginning of the sentence is quadrant 4. We then calculate the quadrant that our prepositional or adjectival noun phrase modifier falls into, and return that value as a feature to our Conditional Random Field model. We divide our sentence into quadrants and ”regularize” our positions so that the magnitude of numerical value of a modifier's position, which happens to be in a very long sentence, does not throw off our model as an outlier.

The reasoning for this theory of ours can be justified as follows. Consider the examples 2.16, 2.17, 2.18, and 2.19 given in Section 2.1. In both the prepositional and adjective noun phrase modifier cases, the non-restrictive modifiers are in the later quadrants. We believe these examples are not just coincidences, as the noun phrase modifiers are later in the sentence, the noun in question is afforded a greater chance to become more definite. As the context that the noun is utilized in becomes clearer, the information that the noun phrase modifier would have otherwise provided to be restrictive information on the modified noun becomes non-restrictive. By that point, the reader already has a clearer idea on the noun, and the noun phrase modifier becomes non-restrictive.

With this logic in mind, we now move into the implementation part of noun phrase restrictiveness classifier models.

5

Implementation of Noun Phrase Restrictiveness Classifiers

In Chapter 5, we first describe the implementation of the methods used in identifying non-restrictive noun phrase modifiers through the methods of Honnibal et al. [12], Dornescu et al. [5], and Stanovsky et al. [26]. We then explain the implementation of our own model that builds on top of the three major papers we have learned and adds its own novel deep semantic analysis methods in distinguishing non-restrictive noun phrase modifiers. The implementation can be found at this GitHub repository.

5.1 Implementing Previous Works

To compare the performance of our own model against those previously detailed in our literature review, we have implemented the algorithms previously discussed in Section 2. We were not able to view and run the exact code written by Honnibal et al. and Dornescu et al. even though we reached out to said authors multiple times. While we were able to view the code written by Stanovsky et al., as multiple libraries they used to train their own models were missing and most-likely compiled by themselves without publishing them on their public GitHub repository, we were not able to directly run Stanovsky et al.'s own code as well. We were also not successful in our attempts to view the libraries used by Stanovsky et al. after multiple contact attempts.

Therefore, we had to replicate all of the previously done models by ourselves from scratch, and benchmark the performance of all these models against our own. These models that we have re-implemented are:

1. Honnibal et al.: Annotating a modifier as restrictive if and only if that modifier is preceded by a comma.

2. Dornescu et al.: Annotating a modifier as restrictive through the supervised CRFsuite classifier over ”standard features used in chunking, such as word form, lemma and part of speech tag”.

3. Stanovsky et al.: Annotating a modifier as restrictive through the supervised CRFsuite classifier by combining the features listed above and introducing new features that they developed on their own such as semantic labelling, the ”name-entity-recognizer” and the ”pre-trained word embedding of the modifier's head words” as detailed in Section 3.3.

5.2 Implementing Deeper Semantic Analyses

The features we have added to our own model are detailed in Chapter 4. The first feature we added, in addition to those of Stanovsky et al. and Dornescu et al., bigram and trigram features that affect its performance set score depending on the popular word-pairings seen with both restrictive and non-restrictive noun phrase modifiers. In particular, our n-gram addition of bigrams and trigrams allow all permutations of two and three adjacent words with their respective features to be considered together. As a result, the relation between adjacent words are given a higher weight. The second feature we added is around inter-sentence distance factor for the location of prepositional and adjectival noun phrase modifiers in sentences. We theorize that the farther an adjectival noun phrase modifier or a prepositional noun phrase modifier is, the more likely it is for such a modifier to be non-restrictive. The reasoning for this is because there are more chances for the noun of the sentence to be clearly defined before the noun phrase modifier is reached.

To train and test both the previously theorized models as well as our own model, we have used the crowdsourced annotated corpus detailed in Section 3.4. We will now view the metric results of both our novel and the previously discussed authors' classifier models.

System	**Feature Type**	**Description**
Honnibal et al.	Preceding comma	w[-1] == ,
Dornescu et al.	Chunking features	Word form, lemma, PoS tags of words above and below current head (separately)
Stanovsky et al. (including Dornescu et al.'s feature)	Enclosing commas	True iff the clause is preceded and terminated with commas
	NER	Tags for *person*, *organization*, and *location*
	Lexical word embeddings	Pre-trained word embeddings from Mikolov et al. [18]
	Modifier type	Modifier type listed in Section 2.1
Our model (including Stanovsky et al.'s features)	Bigrams and Trigrams (N-Grams)	Word form, lemma, PoS and modifier tags of the previous, current and next words, all together in one tuple per each word (together)
	Relative position of Prepositional and Adjectival Noun Phrase Modifiers	The farther Adj. or Prep. modifiers are from the beginning of the sentence, the more likely they are non-restrictive

Figure 5.2.1: Features utilized in each of the models described from literature, as well as in our own model. Note: head refers to the head of the modifier phrase, *separately* refers to the addition of the features separately, *together* refers to the addition of the features as one whole tuple.

6
Results

In Chapter 6, we overview the metric results of the four classifier models discussed in this paper: Honnibal et al.'s simple comma model, Dornescu et al.'s simple machine learning model, Stanovsky et al.'s extensive machine learning model, and our own novel model. We then continue into discussion on the results we have obtained, and provide reasoning behind the results we obtain.

6.1 Observations

In the tables above, we have detailed the performance of our restrictiveness classifier machine learning model with our features, in addition to the performance of the three models from literature detailed in Sections 3.1, 3.2, and

Modifier Type	#	Precision			
		Honnibal et al.	Dornescu et al.	Stanovsky et al.	**Our novel model**
Prepositional	135	.83	.57	.57	**.65**
Adjectival	109	.33	.36	.36	**.35**
Appositive	78	.77	.75	.77	**.82**
Non-Finite	55	.77	.77	.78	**.75**
Verbal	20	0.0	.78	.75	**.75**
Relative clause	13	1.0	1.0	1.0	**1.0**
Total	410	.72	.65	.65	**.68**

Figure 6.1.1: Comparison of the three previous studies' models vs. our own model with the Precision benchmark

Modifier Type	#	Recall			
		Honnibal et al.	Dornescu et al.	Stanovsky et al.	**Our novel model**
Prepositional	135	.10	.38	.38	**.41**
Adjectival	109	.06	.31	.31	**.28**
Appositive	78	.34	1.0	.92	**.86**
Non-Finite	55	.29	.89	.88	**.94**
Verbal	20	0.0	.88	1.0	**1.0**
Relative clause	13	.27	1.0	1.0	**1.0**
Total	410	.19	.62	.63	**.62**

Figure 6.1.2: Comparison of the three previous studies' models vs. our own model with the Recall benchmark

Modifier Type	#	F1			
		Honnibal et al.	Dornescu et al.	Stanovsky et al.	**Our novel model**
Prepositional	135	.18	.45	.45	**.50**
Adjectival	109	.11	.33	.33	**.31**
Appositive	78	.47	.86	.84	**.84**
Non-Finite	55	.42	.82	.82	**.83**
Verbal	20	0.0	.82	.86	**.86**
Relative clause	13	.43	1.0	1.0	**1.0**
Total	410	.30	.64	.64	**.65**

Figure 6.1.3: Comparison of the three previous studies' models vs. our own model with the F-1 benchmark

3.3. For our performance comparisons, we have chosen the Precision, Recall and F1 metrics detailed in Chapter 2.

6.2 Discussion

As seen in the tables above, with the exception of the *precision* metric, our model and the models of Stanovsky et al. and Dornescu et al., are performing better on *recall* and *F1* performance tests than Honnibal et al.'s model. One reason this is the case is Honnibal et al.'s method does not utilize machine learning techniques; rather, it just checks if the sentence article before it is a comma, and labels that modifier as restrictive if it is, and as non-restrictive if it is not.

We have detailed the obtained performance differences of our own classifier machine learning model compared to others previously mentioned below:

1. Our novel model performed better overall with the metrics Precision and F1, while narrowly losing out to the model of Stanovsky et al. in Recall by 0.01.

2. With all metrics, our novel better performed the best while predicting the restrictiveness of prepositional noun phrase modifiers.

3. In terms of adjectival noun phrase modifiers, our novel model barely loses to the models of Stanovsky et al. and Dornescu et al. with Precision and Recall and F1. However, the greatest difference between these is a value of 0.03, as such the difference is not too significant.

While our model is performing better in many metrics compared to the currently best-performing model in literature, which is that of Stanovsky et al., our model is not doing significantly better in all factors. This may be because the features we have added to our own model may not be the best differentiating factor between restrictive and non-restrictive prepositional and adjectival noun phrase modifiers. However, on multiple metrics, our novel machine learning model provides better metrics on certain fronts.

It should be noted that the metric scores that Stanovsky et al. reported in their paper are slightly different from the metric scores we obtained from our reproduction of their code. As stated in Section 5.1, this is partly due to the fact we did not possess the original code repositories of Honnibal et al. and Dornescu et al., and that while we did have access to Stanovsky et al.'s repository, most of their code did not work even after following their Read-me instructions.

As a result, the works of all three papers had to be re-engineered and reproduced. Because of this, minor variations in performance are to be expected. In addition, Stanovsky et al. detailed two features in their paper, ”prepositions”, and ”governing relative”, that were not possible to replicate in our code. No further info apart from mentioning the ”prepositions” feature was given on this feature in the rest of their paper nor in their codebase, and the code provided for replicating the ”governing relative” feature required missing custom-built libraries that did not exist on the internet, and that Stanovsky et al. were not able to provide, even after multiple attempts at reaching out to them. As such, we were not able to replicate two features of Stanovsky et al.'s classifier model, which also contributed to the lower metric scores for their model in our testings.

7 Conclusion

In this thesis, we have devised and implemented a novel and successful method of predicting the restrictiveness of different types of noun phrase modifiers through deep semantic analysis methods. In addition, we brought together the works done in the past in pursuit of identifying non-restrictive noun phrase modifications, and reproduced, thereby verifying the success of these same works. Furthermore, we brought together a literary corpus that includes a wide degree of information, from the metrics for evaluating a classifier model to explaining a fundamental classifier machine learning model, where any person interested in engaging with further research on distinguishing non-restrictive noun phrase modifications can have a head-start from this study.

To get to this level, we first understood what makes noun phrase modifiers restrictive vs. non-restrictive, and realized the importance of obtaining a model that can accurately predict the restrictiveness states of these noun phrase modifiers. We then combed through previous literature on methods of annotating noun phrase modifiers to understand and observe what makes a good corpus of noun phrase modifiers, and which distinguishing methods were the most rewarding in terms of prediction success, before implementing three key methods of predicting non-restrictive noun phrase modifiers to check the authors' claims of accuracy and correctness. We then theorized new deep semantic analysis techniques that can help us achieve better scores for predicting and annotating non-restrictive noun phrase modifications and implemented our own novel model that built upon both previous literature and our own contributions. While our final results show that our model does not perform significantly better than that of Stanovsky et al., it does indeed show promise in the payoffs that deep semantic analysis can provide.

The field of predicting non-restrictive noun phrase modifications is an ever-enlarging area of study. With continued research in the field, which are supported by new Natural Language Processing and Computational Semantic tools that continue to come out, academic research in this classifier machine learning area is bound to increase.

Bibliography

[1] Diti Bhadra. Lecture notes in semantic theory i, March 2018.

[2] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[3] Xavier Carreras and Lluís Màrquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, pages 152–164, 2005.

[4] Robert S Cooper, Jeff F McElroy, Walter Rolandi, Derek Sanders, and Richard M Ulmer. Personal virtual assistant with semantic tagging, October 15 2002. US Patent 6,466,654.

[5] Iustin Dornescu, Richard Evans, and Constantin Orasan. Relative clause extraction for syntactic simplification. In *Proceedings of the Workshop on Automatic Text Simplification-Methods and Applications in the Multilingual Society (ATS-MA 2014)*, pages 1–10, 2014.

[6] Markus Egg. The syntax and semantics of relative clause modification. *Proceedings of the Sixteenth Computational Linguistics in the Netherlands*, pages 49–56, 2007.

[7] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[8] Joseph L Fleiss and Jacob Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619, 1973.

[9] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. 2010.

[10] Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics, 2009.

[11] Julia Hockenmaier and Mark Steedman. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396, 2007.

[12] Matthew Honnibal, James R Curran, and Johan Bos. Rebanking ccgbank for improved np interpretation. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 207–215. Association for Computational Linguistics, 2010.

[13] Rodnry Huddleston and Geqffrry Pullum. The cambridge grammar of the english language. *Zeitschrift für Anglistik und Amerikanistik*, 53(2):193–194, 2005.

[14] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[15] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[16] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

[17] Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The nombank project: An

interim report. In *Proceedings of the workshop frontiers in corpus annotation at hlt-naacl 2004*, 2004.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[19] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs). 2007.

[20] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1): 71–106, 2005.

[21] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[22] Ron Rower. Restrictive and non-restrictive modifiers, May 2013. URL http://www.grammar-once-and-for-all.com/punctuation/restrictive-and-non-restrictive-modifiers/.

[23] Sunita Sarawagi and William W Cohen. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192, 2005.

[24] Motoki Sato, Hiroyuki Shindo, Ikuya Yamada, and Yuji Matsumoto. Segment-level neural conditional random fields for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 97–102, 2017.

[25] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.

[26] Gabriel Stanovsky and Ido Dagan. Annotating and predicting non-restrictive noun phrase modifications. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1256–1265, 2016.

[27] Hanna M Wallach. Conditional random fields: An introduction. *Technical Reports (CIS)*, page 22, 2004.