InfoCorpus: Generalising Hedge-Related Tasks Across Scientific Domains

Daniel Kim

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

Researchers enhance the credibility of their claims and analyses through various approaches. One such method is with hedges. Hedges are single or multi-word expressions that express uncertainty. Some examples are: *'may'* or *'whether'*. These hedges also exhibit certain scopes, which are clauses affected by the hedge. To this end, we examine the detection of these linguistic devices using state-of-the-art machine-learning models. We also hypothesise that their use transcends scientific domains. I.e., they are domain-independent.

First, we briefly examine previous work on their detection, which varies in complexity from a simple bag of words approach to advanced neural networks. Section three reproduces and expands on previous work using state-of-the-art transformer models to detect hedge cues and resolve hedge scopes. We achieve around 0.80 F1 score, similar to previous work. Although unrealistic, we also show an upper bound of these models by stratifying the data.

Furthermore, we present a new hedge dataset in section four. We improve upon BioScope and WikiWeasel, the two main data sets used for hedge detection, by introducing *InfoCorpus*. This data set contains annotations about hedge cues, their scopes, and uncertainty types, the novel combination of which has never been seen in a single dataset. We analyse the cue and scope distributions in this dataset and find that they are similar to those of BioScope. This implies that the use of hedging is alike regardless of the domain (Biomedical and Informatics). Consequently, this dataset allows us to answer our main hypothesis - 'Can we generalise hedge cue detection and scope resolution?' which we answer in section five.

In section five, we perform domain adaptation to show that we can generalise hedge cue detection and scope resolution to more scientific domains. We train the best-performing models on BioScope and test them on InfoCorpus. We achieve a slightly lower F1 score of around 0.70 F1 score. From this, we find that indeed the model performs to a similar or slightly lower calibre. This suggests that hedge cues may not be domain-dependent. An interesting question for future work is to investigate whether certain hedges and the preference to use one uncertainty type over another depend on an author's writing style. Furthermore, future work should expand on both the annotation guidelines for InfoCorpus and its contents.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 648491
Date when approval was obtained: 2024-12-20

The participants' information sheet and a consent form are included in the appendix.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Daniel Kim*)

Table of Contents

1 Introduction **1**
- 1.1 Motivations . . . 1
- 1.2 Report Structure . . . 2

2 Background Review **4**
- 2.1 Hedge Cues . . . 4
  - 2.1.1 Epistemic Uncertainty . . . 5
  - 2.1.2 Hypothetical Uncertainty . . . 5
- 2.2 Hedge Scopes . . . 6
- 2.3 Hedge Datasets . . . 6
  - 2.3.1 BioScope corpus . . . 6
  - 2.3.2 WikiWeasel . . . 7
- 2.4 Evaluation Metrics and Terminology . . . 7
  - 2.4.1 Precision, Recall, F1 Score . . . 7
  - 2.4.2 Accuracy . . . 8
  - 2.4.3 Left-Hand Scopes, Right-Hand Scopes . . . 8
  - 2.4.4 Cohen's Kappa . . . 8
- 2.5 Recent Work . . . 9
  - 2.5.1 Early Work . . . 9
  - 2.5.2 CoNLL 2010 Shared Task . . . 9
  - 2.5.3 Transformers . . . 10
  - 2.5.4 Methodologies . . . 11
- 2.6 In conclusion . . . 11

3 Reproduction of Previous Work **12**
- 3.1 Methodology . . . 12
  - 3.1.1 Preprocessing . . . 12
  - 3.1.2 Dataset Splitting . . . 12
  - 3.1.3 Class Weighting . . . 13
  - 3.1.4 Token Aggregation Methods . . . 14
  - 3.1.5 Hedge Cue Detection . . . 14
  - 3.1.6 Hedge Scope Resolution . . . 14
- 3.2 Results . . . 15
  - 3.2.1 Hedge Cue Detection . . . 15
  - 3.2.2 Hedge Scope Resolution . . . 15

3.3 Discussion . . . 16
3.3.1 Hedge Cue Detection . . . 16
3.3.2 Hedge Scope Resolution . . . 17
3.4 Conclusion . . . 19

4 InfoCorpus **20**
4.1 Overview . . . 20
4.2 Annotation Methodology . . . 21
4.2.1 Cue . . . 21
4.2.2 Scope . . . 23
4.3 Results and Discussion . . . 25
4.3.1 Cue annotations . . . 25
4.3.2 Scope annotations . . . 27
4.4 Comparisons to BioScope and WikiWeasel . . . 29
4.5 In conclusion . . . 30

5 In-Domain and Domain-Adapted Tasks on InfoCorpus **32**
5.1 Methodology . . . 32
5.2 Same Domain Results . . . 32
5.2.1 Hedge Cue Detection . . . 32
5.2.2 Hedge Cue Type Detection . . . 33
5.2.3 Hedge Scope Resolution . . . 33
5.3 Domain Adaptation Results . . . 33
5.3.1 Hedge Cue Detection . . . 34
5.3.2 Hedge Scope Resolution . . . 34
5.4 Discussion . . . 34
5.4.1 Validation for dataset stratification . . . 34
5.4.2 Cue Detection . . . 35
5.4.3 Type Detection . . . 36
5.4.4 Scope Resolution . . . 37
5.5 Conclusion . . . 38

6 Conclusion **39**
6.1 Reproduction of Previous Work . . . 39
6.2 InfoCorpus . . . 39
6.3 Domain Adaptation . . . 40
6.4 Future Work . . . 40

A Byte Pair Encoding **44**
A.1 Example of Byte Pair Encoding . . . 44
A.2 'Solving' the tokenisation problem . . . 44

B False Positives and Negatives of various models **45**
B.1 False positives and false Negatives for non-stratified data and stratified data (BioScope) on BERT . . . 45
B.2 False positives and negatives of BERT for hedge cue type detection (InfoCorpus) . . . 47

B.3 False positives and negatives for domain-adapted and in-domain BERT (InfoCorpus) 48
B.4 False positives and false negatives for in-domain RoBERTa for hedge cue detection (InfoCorpus) 49

C InfoCorpus Pipeline **50**

D Participation Information Sheet **51**

E Participation Consent Sheet **55**

Chapter 1

Introduction

1.1 Motivations

Linguistic devices and phenomena have been the spotlight of studies for as long as they have existed. These range from simple devices such as metaphors and similes, to more complex ones, like euphemisms and named entities. As such, we have also poured our efforts into detecting them through various means. One major front is the CoNLL (Conference on Computational Natural Language Learning), where researchers aim to solve certain natural language tasks. Some previous tasks were: grammatical error correction [28], semantic role labelling [6], and more relevant to our project, hedge detection [7].

Hedges, or hedging, weaken the 'directness' of our statements. For example, take the sentences: *'It is raining'* and *'It **may** be raining'*. In the first instance, we are confident in our statement, that it is indeed raining and there is no question about it. In contrast, the hedge cue *'may'* in the latter sentence introduces uncertainty into the sentence, thus reducing the strength or directness of the claim. We will be investigating two main types of uncertainty: hypothetical and epistemic.

Researchers and linguists believe that in moderation, this linguistic device lends more credibility to our findings [16] and that hedges "constitute an essential element of argumentation" [17]. Its overuse may lead to distrust in the evaluations of the authors, while the opposite could show an overstatement of the results. We give another example sentence pair: *'The model is overfitting'* and *'It seems somewhat possible, that the model might be, to some extent, overfitting'*. The author of the first sentence is confident about their findings - perhaps a little too confident and terse. Therefore, readers might be less prone to believe the findings. Rarely are results definitive in scientific research. Similarly, we are immediately put off by the claim in the second sentence. Heavy hedging not only bloats the sentence but also leads the readers to question the validity of the statement.

Researchers have employed multiple ways to detect these phenomena to varying degrees of success. These methods range from simple bag of words (not accounting for the relative positions of words) to more complicated neural networks. In this paper, we show

that the state-of-the-art machine learning models excel at hedge detection by reproducing previous work. Moreover, we primarily investigate whether the use and types of hedging are common across all scientific domains (e.g. Biomedical, Informatics). We hypothesise that hedges are domain-independent. Some hedge words are only used for hedging, such as **may**, whereas others display more ambiguous behaviours, like *could*. Therefore, we employ domain adaptation, which involves testing models on a different domain from that they were trained on. Through this approach, we aim to either confirm or deny this hypothesis.

To this end, we investigate the following claims:

1. Can we successfully reproduce previous work on hedge cue detection and scope resolution?
2. Can we maintain a consistent score while varying the distribution of the data for cue detection?
3. How difficult is it to identify hedge cues and scopes?
4. Can we generalise hedge cue detection and scope resolution to more scientific domains?

Next, we outline where we answer these questions and briefly summarise our findings.

1.2 Report Structure

We structure our report as such: first, we set up some background information regarding hedges and their scopes. We introduce various evaluation metrics that are used throughout the paper, and we examine previous work that has led to the current state-of-the-art.

Next, we answer the first two claims in section three. We investigate whether transformer models can perform hedge cue detection and scope resolution on the BioScope corpus. Previous work has accomplished the best score of 0.83 F1 score (using XLNet [40]), while we show that our models also achieve a similar score. We prove our second claim in section five but show that it is possible in section three. We briefly outline a stratified sampling method, which results in models that exhibit the same score while training on fewer data. However, this is only possible in a lab setting where the distribution of data is known (including the test set). For scope resolution, we successfully reproduce previous work. We show the results and analyse errors to better understand the capabilities of these models.

We then spend some time analysing our third claim by examining the annotation guidelines. We note that although some hedge cues are explicit (e.g. *'may'*), others require disambiguation of the context (e.g. *'can'*). We were only able to aggregate annotations for hedge cues (with a Cohen's Kappa score of 0.92), as the other annotator faced scheduling issues. However, we also verified that hedge scopes were systematic. Therefore, it was simple enough to label the scope as the RHS (Right Hand Scope) from the cue word (if it was either a verb or an auxiliary) to the delimiter.

We compile these annotations into **InfoCorpus**. This corpus contains sentences from scientific papers in the Informatics domain (more specifically, Computational Linguistics). During its annotation process, we encounter difficulties, particularly in the cue annotation phase. This is in line with the model results, as cue detection appears to be a harder task than scope resolution. Through this novel dataset, we aim to test whether domain adaptation is possible in section five. We conduct comparisons between BioScope and InfoCorpus in terms of their hedge distribution. We also see similar distributions of hedge types and hedge scopes across the corpora.

InfoCorpus contains annotations regarding cue types (epistemic or hypothetical). This, in turn, makes our corpus the first (to the best of our knowledge) to include all three annotation types (cue, cue type, and scopes). This was instrumental in performing additional tasks in the next section.

In section five, we investigate our main hypothesis by performing in-domain and domain-adapted tasks. This answers our fourth and main claim. Furthermore, we add hedge-type classification to the suite of tasks at hand, which achieves the same score as the other tasks (0.84 F1 score). Most of our domain-adapted models performed at a similar level to in-domain models (0.73 and 0.91 F1 scores for cue detection and scope resolution respectively). Therefore, we arrive at a result that suggests that we can generalise hedge cues and scopes to more scientific domains.

Finally, we conclude our report by summarising our findings. We also include some areas of further improvement, such as expanding the InfoCorpus dataset, tightening the annotation guidelines, and investigating whether different writing styles affect the usage of certain hedges and hedge types.

Chapter 2

Background Review

2.1 Hedge Cues

Authors see hedges as a linguistic device which shows their certainty. This term was originally coined by Lakoff [23], where he describes them as "words whose meanings implicitly involve fuzziness". For example, let us explore the sentence *"A dog is sort of a fish"*. The hedge *'sort of'* takes the subject *'dog'* and fuzzes its meaning, and thus certainty, into the object *'fish'*. Other examples of such hedge words are *'almost'*, *'possible'*, and *'may'*.

More recently, William [22] writes that hedges are a necessary linguistic tool to convey an author's certainty in writing. When used liberally, a passage seems too vague, whereas when hardly used, readers have a hard time trusting authors with their blunt assertions. Thus, he concludes that scientific authors must use this device sparingly. Moreover, he explains that "if you state a claim moderately, readers are more likely to consider it thoughtfully". All scientific authors want readers to appreciate their findings, and hedges are an important stepping stone towards achieving their goal.

We see that multiple corpora have been annotated with uncertainty cues 2.1. However,

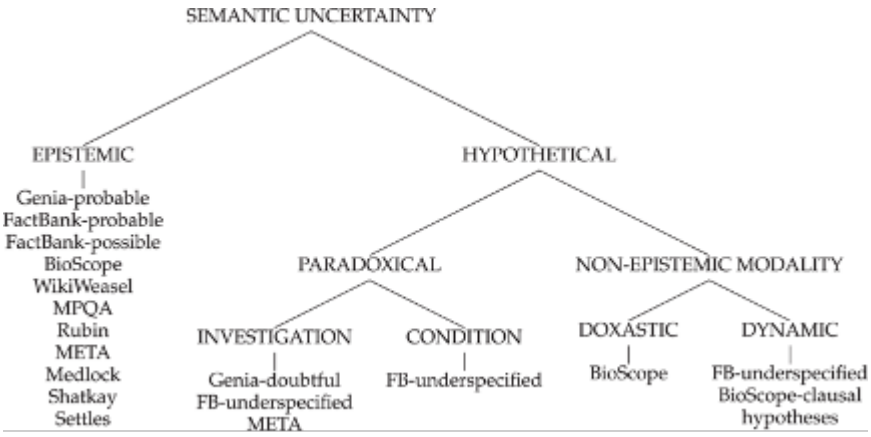


Figure 2.1: Hierarchical view of the different uncertainty types.

we only focus on the BioScope and WikiWeasel as researchers have published studies on them. We also see that there are numerous sub-categories of uncertainty, but we limit the scope of our paper to the top-most layer - *epistemic* and *hypothetical* uncertainty.

2.1.1 Epistemic Uncertainty

We follow the definition of *epistemic* in the Oxford dictionary [32]: *"relating to knowledge or to the degree of its validation"*. Thus, we classify epistemic uncertainty as uncertainty which shows a lack of knowledge about a state or an event. I.e., given an event, our knowledge of it cannot be certain whether it is true or false. This definition is also echoed by Szarvas et al [38]. Here is an example of epistemic uncertainty where the uncertainty cue is bolded:

*This assignment **could** be due next week.*

The subject knows that the assignment is indeed due, but they do not know *when* it is due. It could be due next week, next month, or even next year. However, this is not certain. An intuitive way to understand epistemic uncertainty is by labelling it as a *known unknown*. A claim could be true or false, and one of them has to be true.

2.1.2 Hypothetical Uncertainty

On the other hand, we define *hypothetical* uncertainty as such: hypothetical uncertainty depicts a world where an event is not certain, similar to epistemic uncertainty. Furthermore, we are also uncertain about our uncertainty (thus, an *unknown unknown*. That is, we do not and cannot deduce the probability of said event being true or false. Here is an example of hypothetical uncertainty, where the uncertainty cue is again, bolded.

*We **speculate** that this sentence contains a hedge cue.*

The word ***speculate*** shows the author's uncertainty about the following clause *'that this sentence contains a hedge cue'*. The author does not know the truth value of the statement 'this sentence contains a hedge cue', and they further are unsure of the probability of the said event occurring.

Moreover, this type of uncertainty (named 'investigative') is a subset of the broader category that is hypothetical. Other aspects include the different types of modality - dynamic and doxastic. Dynamic modality refers to events in the future, while doxastic modality references a speaker's beliefs. Finally, conditional clauses also express hypothetical uncertainty when used in the present case. Examples of these three cases can be seen below:

Dynamic: *I **have** to do my assignment.*
Doxastic: *I **believe** that my assignment is due soon.*
Conditional: ***If** I leave, I can't do my assignment.*

For this paper, we have combined these sub-categories into a general *hypothetical uncertainty* label to avoid sparsity of data when annotating.

We must draw a boundary between linguistic uncertainty and uncertainty used in a

machine learning concept. Epistemic and aleatoric uncertainty [15] in machine learning deals with a model's predictions. When a model's decision is considered 'epistemic', we associate this with the lack of knowledge of the perfect predictor. I.e., either the model has never seen that kind of input before, and even if it has, the input itself is ambiguous to produce a perfectly confident answer (e.g. part of speech disambiguation only given the word). On the other hand, we can classify aleatoric uncertainty as caused by the stochastic nature of machine learning models (e.g. calculating the conditional probability $P(y \mid x)$). Although epistemic uncertainty in both scenarios is quite similar, we would like to clarify that we are solely interested in linguistic uncertainties, i.e., hypothetical and epistemic uncertainty.

2.2 Hedge Scopes

Hedge scopes are clauses that the respective hedge cues affect, and thus lower the confidence of the semantics of said clauses. We use braces to show the beginning and end of hedge scopes ({ and } respectively). Looking at the sentences below, the cue **raises the possibility** affects the meaning of the sentence, and therefore, its scope covers the entire sentence.

This { ***raises the possibility*** *that adenosine plays a role in the control of metamorphosis as well as in the response to stress* } .

We describe hedge cues and hedge scopes in more detail in section four.

2.3 Hedge Datasets

2.3.1 BioScope corpus

The BioScope corpus [38] is comprised of biomedical papers which have been annotated for negation and hedges. Sentences that contain these linguistic phenomena have the following qualities: the word(s) is labelled with its corresponding attribute (i.e., *negation* or *speculation*) and cue ID (E.g. for a cue in paper 2 sentence 20, cues found in this sentence would be labelled with the ID X2.20.1, X2.20.2, etc); for each cue in the sentence, the corpus also contains their respective scopes labelled with the cue's ID.

Auxiliaries (E)	Verbs (H, E)	Adjectives/Adverbs (E)	Conjunctions (H, E)
May	Suggest	Probable	or
Might	Question	Likely	either..or
Can	Presume	Possible	... versus ...
Could	Suspect	Unsure	and/or
Would	Indicate that	Unlikely	whether
Should	Suppose		
	Seem		
	Appear		

Table 2.1: Hedge Cue Examples

The table below 2.1 consists of the uncertainty types and some example cues. We see words such as *'may'* and *'might'* always annotated as a cue. On the other hand, other more 'general' words such as *'or'* or *'can'* depend on the context that they appear in. This came to be a major challenge that we had to overcome during our annotation of InfoCorpus.

2.3.2 WikiWeasel

WikiWeasel [14] is a dataset that contains `Weasel` [8] tags found in Wikipedia articles. These tags are matched to words which suggest a degree of uncertainty, which must be removed to preserve the factual nature of Wikipedia articles. The layout of this dataset is similar to that of BioScope, in that each sentence is labelled with a sentence ID and may contain at least one hedge cue.

Key differences to the BioScope corpus include the uncertainty type within each hedge cue (i.e. hypothetical conditional, hypothetical doxastic, modal probable). There are two limitations to this dataset. Firstly, the data set does not contain any hedge scopes, so it is only possible to perform hedge cue detection. Moreover, weasel tags are temporary and are removed once authors rewrite the article. Thus, it is difficult to check existing articles as they have been rewritten to omit the weasel tags.

2.4 Evaluation Metrics and Terminology

We use the following metrics and methods to evaluate the performance of models and annotations. These metrics are specialised to be more sensitive to some aspects of the results than others and are used depending on the context.

We consider these evaluation metrics at the token-ID level. We elaborate on token IDs in the 'transformer' section. This means that for multi-word expressions, each ID within that word would be assigned a label. Although the initial worry was that IDs in multi-word expressions would be assigned different labels, we found that this was not the case in actuality.

2.4.1 Precision, Recall, F1 Score

Precision measures a model's ability to assign the correct label for a given prediction. The formula is given below:

$$Pr = \frac{TP}{TP + FP} \tag{2.1}$$

On the other hand, recall measures a model's ability to correctly return one class' labels for its predictions. Precision and recall are inversely proportional to each other. Thus, if we achieve a 1.00 precision by not classifying anything ($TP = 0, TP + FP = 0$), the model's recall is consequently 0.00.

$$R = \frac{TP}{TP + FN} \tag{2.2}$$

To address this issue, we use the F1 score to provide a single evaluation metric for a model's performance. An F1 score takes both precision and recall into account, taking their harmonic mean. This allows extreme values to be reflected in the final value, whereas if we had used a simple average $\frac{Pr+R}{2}$, the extremes would have been smoothed out.

$$F1 = \frac{2 * (Pr * R)}{Pr + R} \tag{2.3}$$

2.4.2 Accuracy

Accuracy is the most basic metric that can be used. Although it is straightforward to interpret, it also could lead to misinterpretations. In the context of hedge detection, there is a huge imbalance in data. For example, in the BioScope dataset, there are a total of 20,000 sentences, and only 10% of them contain at least one hedge cue. If we calculate a model's accuracy in predicting a rare class C, its accuracy would still be incredibly high even if the model did not predict anything correctly (since $TN \gg TP$). Thus, we do not consider accuracy to be a valid metric to use in our paper.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{2.4}$$

2.4.3 Left-Hand Scopes, Right-Hand Scopes

These two sub-clauses pertain to the scopes of hedge cues. LHS (Left-Hand Scopes) deal with clauses that are to the left of a given cue word and vice versa for RHS (Right-Hand Scopes).

2.4.4 Cohen's Kappa

We use Cohen's Kappa values to evaluate annotations. This value calculates a finer-grained value by taking random chance into account. For cues, we take the individual word as is given. On the other hand, we use the aforementioned LHS, RHS, and FS chunks to determine the quality of annotations. To calculate Cohen's Kappa, we use the following formulae 2.5. $P(O)$ stands for the observed probability. I.e., the number of agreements divided by the total number of gold labels. $P(E)$ is the expected probability, which is the number of gold labels divided by the total number of items in the collection.

$$\kappa = \frac{P(O) - P(E)}{1 - P(E)} \tag{2.5}$$

2.5 Recent Work

We look back at recent work performed on hedge cue detection and scope resolution below. These methodologies range from a simple bag of words approach to complex neural networks with varying results. For our paper, we focus primarily on the state-of-the-art approach but describe earlier works which have influenced where we stand today.

2.5.1 Early Work

We briefly introduce some early methods that researchers employed. They have not been tested on the BioScope or WikiWeasel dataset, so there are no grounds for comparisons against more recent work.

Light et al [24] proposes two methods: string-matching and SVM. Kilcoglu and Bergler [21] extend their work by incorporating WordNet [13] into a weakly supervised probabilistic model. This expands the hedge word dictionary which allows them to analyse syntactic patterns for hedges. Furthermore, they mention that future improvements could be made. For example, more negative quantifiers could be added to the dictionary, such as *'little'*, which produces a hedge when combined with non-speculative such as *'little was known'*. Another improvement to this method could incorporate word embeddings such as GloVe [31]. WordNet operates at a thesaurus level and does not consider the semantics of each token as is. Therefore, we could perform more deliberate expansions using word vector embeddings to calculate whether the semantics of the word are related to hedges.

Szarvas [33] follows on from Medlock and Briscoe [26] by using a maximum entropy model. They suggest that some words, especially stop words, are also classified as identifiers for hedges. A good example is the word *it*, which is used for many hedge verbs such as *it appears*, *it shows*, *it is likely*, and more. To combat the noise, they introduced bi-grams and tri-grams into the probability model and filtered out any features which were sub-strings of a longer string.

2.5.2 CoNLL 2010 Shared Task

More recently, researchers participated in the CoNLL-2010 Shared Task [7]. They were tasked with answering two questions: detecting uncertainty in sentences, and measuring their scope. They were given two data sets, which were also used in previous papers mentioned above: BioScope and Wikipedia articles containing weasel tags. Here, we summarise their findings and also their limitations.

On the BioScope data set, papers reported results ranging from around 30.3 to 86.4. However, an important distinction is that they carried out a classification problem (e.g. whether sentences contained at least one cue) as opposed to performing token classification (e.g. whether each token in the sentence is a hedge). Furthermore, these papers mostly agree that a binary classification of uncertainty in sentences might not be the best way to map complex sentences that might contain more than one hedge [34].

What differs between each paper is their approach to detecting hedge cues and their scope. The most common machine learning models appear to be an ensemble method with CRF (Conditional Random Fields) and MaxEnt classifiers. There are some exceptions, however. Kilicoglu and Bergler [20] use a rule-based method which extends their previous work [21].

Ji et al [18] proposes a system using Average Perceptron. Their implementation ranked the highest in terms of precision (0.942), but the lowest in recall (0.066). They employ n-gram patterns which are used as features. There is room for improvement here. Ji et al mention that they would like to experiment with other features such as chunks. Moreover, we could use more advanced algorithms than the Average Perceptron, such as a neural network.

Examining the scope resolution task, we again see similarities. Most papers use a Conditional Random Field, although Kilicoglu and Bergler [20] stick with their rule-based methods. Farkas [12] also points out that the features used in these models were also very similar to those used for hedge classification, with the addition of dependency relationships between phrases and hedge cues.

Morante et al [27] uses a kNN memory-based model *TiMBL* [9], As a result, they achieved the highest F1 score of 0.57. For the model's features, they used PoS (Part of Speech) tags, clause construction, and other features extracted from the dependency trees. There is much work to be done in hedge scope resolution. They also advise using different machine learning techniques, specifically on identifying the scope of hedge cues. We intend to investigate whether current state-of-the-art models can perform better.

2.5.3 Transformers

Before we describe the most recent works in hedge detection, we must understand the state-of-the-art model architecture used in these papers. BERT is pre-trained on a large corpus which varies for each model. Afterwards, BERT can be fine-tuned on a different dataset for a variety of tasks such as Named Entity Recognition, Question Answering, and Token Classification.BERT takes in two types of data as input: Token IDs and Attention Masks

Token IDs relate to the mapping between each word to one or more tokens. The original BERT uses WordPiece embeddings developed by Wu et al [39] which contains a vocabulary consisting of 30,000 tokens. Moreover, the input sentence must be of the same length. This is achieved by concatenating `[PAD]` tokens to the end of each sentence. We show an example of tokenisation in A.1 without padding.

Some models, such as RoBERTa [25], perform byte-pair tokenisation. BPE (Byte-Pair Encoding) is an algorithm which compresses text encodings and allows for better generalizability of tokenization in various domains. For example, BERT would tokenize the right bracket and full stop (*).*) as two separate token IDS (1007, 1012), whereas RoBERTa would compress these encodings into one token ID (322). This difference proved to be tricky when pre-processing our dataset. Furthermore, we believe that BPE

introduced problems when training RoBERTa, leading to erroneous and inexplicable results seen in later sections.

2.5.4 Methodologies

Britto and Khandelwal [5] propose training and testing transformers on the BioScope dataset and the SFU review dataset. We only look at the former (BioScope) as it contains scientific papers, whereas the former contains reviews for different products. Using three BERT variants (BERT, XLNet [40], RoBERTa), they achieve significant improvements on both tasks. They define their classes as being the following: (Non-Cue, Single-Word-Cue, Multi-Word-Cue, and Padding). They fine-tune each model on both the BioScope Abstracts (BA) and BioScope Full Papers (BF), after which they perform domain adaptation or joint training to analyse the models' generalizability.

To compare our findings, we examine the models that have been trained and tested only on the full Bioscope papers (BF) 2.2. Moreover, this table includes notable previous works. We could not include some papers that we included earlier as they performed slightly different tasks (sentence-level vs word-level hedge detection).

Papers	**F1**
BERT [5]	81.66
XLNet [5]	83.24
RoBERTa [5]	79.31
Tang et al [35]	81.3
Velldal et al [37]	78.7
Ji et al [19]	77.44

(a) Hedge Cue Detection Performances

Papers	**F1 (Average)**	**F1 (First)**
BERT [5]	91.60	90.42
XLNet [5]	83.24	91.88
RoBERTa [5]	92.02	91.30
Morante [27]	57.32	-
Kilicoglu [20]	55.21	-

(b) Hedge Scope Resolution Performances

Table 2.2: Hedge Detection and Scope Resolution Performances

2.6 In conclusion

Researchers faced several limitations when completing these challenges: lack of annotated data and old methodologies. The state-of-the-art implementation only showed that scope resolution was possible in the biomedical domain, and earlier works did not perform well at all. We aim to re-implement previous work with improvements in both pre-processing and model training to obtain a better result. Moreover, we will generalise this task to more domains in the form of *InfoCorpus*, which contains papers from Computational Linguistics.

In the rest of this paper, we first reproduce previous state-of-the-art hedge cue and scope detection using BERT and its variants. Next, we introduce *InfoCorpus*: a dataset containing hedge cues and their scopes in the same style as *BioScope* and *WikiWeasel*. We compare these three datasets, noting similarities and differences in annotated cues. Finally, we train and test BERT variants through domain adaptation and discuss the results.

Chapter 3

Reproduction of Previous Work

In this section, we perform cue detection and scope resolution on the BioScope dataset. Our methodology stems from previous work, but we also incorporate our approaches to reinforce areas where the previous paper was unclear. We omit results on the Wikiweasel dataset due to the huge amount of resources required to train BERT variants for 60 epochs. We first show the pre-processing steps of the BioScope data. Then, we perform hedge cue detection followed by hedge scope resolution. We achieve a similar performance while varying the distribution of the training data, which suggests that we can use less data to train these models. However, we note that this is unrealistic. We finally present our results and discuss them.

3.1 Methodology

3.1.1 Preprocessing

We struggled to extract relevant information from the BioScope XML file. Due to their file structure, we encountered difficulties matching cues and their scopes. Therefore, we used multiple sliding windows to generate class labels for each token. Although largely successful, we found that this algorithm misclassified certain words if they contained a cue within them. For example, we show that **StratifiedBERT (1-to-4)** misclassified *'orthology'* B.1b as not a cue when in fact we labelled it as a hedge. This was due to the presence of the hedge cue 'or' in the same sentence. Thus, *'orthology'* was also mislabeled as a hedge. However, this was an isolated instance and we have confirmed that the algorithm has not mistakenly labelled other words as cues.

3.1.2 Dataset Splitting

We also explore some dataset-splitting methods. Current state-of-the-art methods do not mention the highly imbalanced labels. Therefore, when performing the standard 70-15-15 splits for training, validation, and test dataset, the resulting dataset could end up with more sentences containing single and multi-cue words than others, leading to two potential outcomes:

- Outcome 1: There are more cue-word-containing sentences in the training set than in the test. Since our models already achieve a near-perfect score for non-cue words, this would result in a higher-than-normal average macro F1 score when testing our model.
- Outcome 2: There are more cue-word-containing sentences in the test set than in the training. The model would not train on any meaningful data. Therefore, we would see the model underfitting to the data and receive a lower-than-expected average macro F1 score during testing.

To avoid these outcomes, we perform a stratified data sampling method. We ensure that each dataset contains the same ratio of sentences that contain at least one hedge cue to those that do not contain any cues. In practice, we cannot guarantee that the proportions of each relevant class (non-cue, single-cue, multi-cue) are the same in the train, validation, and test set. Furthermore, we did not see any noticeable improvements when using this stratified method compared to previous work.

Moreover, we perform another set of stratified sampling. This time, we balance the proportion of sentences that contain hedge cues to those that do not on a 1-to-k basis (where k is the proportion of sentences that are hedge-free). We found that not only did this approach reduce training times, but it also maintained a relatively consistent F1 score. We validate this behaviour in section five. Of course, we must also remember that this provides an optimistic upper bound on the score. This behaviour is impossible to recreate in a realistic scenario and was only a test to see the potential of these models.

Future work could examine the number of hedge cues present in each sentence. Currently, we do not take this into account and only flag a sentence as either containing at least one cue or not. Therefore, potential improvements could include adding weights to each sentence based on the number of cues. However, we must note that this is purely in an experimental setting and thus not reflective of realistic distributions of class labels (we cannot know this before examining the entire dataset).

3.1.3 Class Weighting

State-of-the-art works also mention that they used class weights to ensure that the model did not train on padding data. They set all weights to 1 except for the padding, which they set to zero. During reproduction, we found that the models' loss would increase exponentially, which suggests that the model was not learning at all from the given data. When inspecting the test results, it would label all tokens as non-cue (including the padding). Therefore, we first omitted these weights from the model parameters. This produced a slightly better result - the model received a 1.00 F1 score for both non-cue words and padding but still struggled when it came to the hedge cues. Instead, we use `sklearn`'s [29] `compute_class_weight` function to learn the weights of each class. This function allocates weights according to the inverse frequency of the class in the data.

3.1.4 Token Aggregation Methods

Britto and Khandelwal introduce two algorithms which aggregate token-level classification into word-level classification. The *Average Token* method takes the sum of probabilities for each token and calculates the most likely label. On the other hand, the *First Token* method assigns labels based on the first token and discards subsequent probabilities for each word. Although these algorithms are used in scope resolution, we also extend them to cue detection. However, we do not see any meaningful increases in the F1 score of the models when compared to the token-ID-level classification.

3.1.5 Hedge Cue Detection

We reproduce previous work done by Britto and Khandelwal. Due to the lack of any meaningful code presented in the paper, we cannot be completely certain that our pre-processing of the BioScope paper matches theirs. However, we *can* confirm that our trained models achieved a similar evaluation score to Britto and Khandelwal. We used a combination of our local machines and Google Colaboratory [4] to train and test our models. We found that a major roadblock was the lack of resources needed to train these models, especially bigger models like XLNet which took around 90 minutes per epoch using a T4 GPU. Therefore, it was imperative to make sure we saved the models' weights to ensure that valuable training time was lost and that we kept these models for domain adaptation, which we elaborate more in section five. Outside of this, we kept all hyper-parameters the same (learning rate = $3e-05$, batch size = 8, epochs=60).

3.1.6 Hedge Scope Resolution

We perform similar steps to that of hedge cue detection for hedge scope resolution. We only tested models on the BioScope data set as WikiWeasel did not contain any scope annotations. Notably, we came across some hurdles when training RoBERTa. As mentioned before, this was due to RoBERTa's byte-pair tokenization, which tokenizes some pairs of characters into a single token ID rather than multiple. Thus, the token IDs of a scope within a sentence contained tokens which were unseen in the sentence. In the sentence below, we see that the end bracket token ')' is included within the scope, while the full stop is not. Therefore, BPE concatenates these two token IDs (43, 4) into a single sub-word token ID (322) denoting an end bracket and full stop (').').

... { **might** provide additional stability to the mature DCC (MatDCC) } .

Fancellu et al. [11] state that punctuations and other delimiters facilitate easier scope resolution for negation (and by extrapolating, for uncertainty). Thus, removing these problematic pairs was not an option. Had it been just the full stop that was causing problems, we could have replaced them with `EOS` (End Of Sentence) tokens, but this was not the case. Therefore, we inserted a blank space between the two characters. Table A.1 shows the before and after of these problematic tokens.

3.2 Results

3.2.1 Hedge Cue Detection

We show the average macro Precision/Recall/F1 score of models (BERT, XLNet, RoBERTa) trained and tested on BioScope. We used early stopping to tune our models. Early stopping ensures that our model does not overfit the training data by finishing the training early if the tracked metric does not increase. Therefore, we track the average macro F1 score and set the patience to six.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.73/0.93/0.80	0.74/0.93/0.81	0.74/0.93/0.81
XLNet	0.72/0.90/0.78	0.72/0.90/0.78	0.72/0.90/0.78
RoBERTa	0.73/0.93/0.80	0.73/0.93/0.80	0.73/0.93/0.80

Table 3.1: Hedge Cue Detection results for BERT, XLNet, and RoBERTa.

We conducted additional experiments to see whether varying the distribution of sentences with hedges in the dataset affected the score. We calculated the total number of sentences with at least one hedge cue (328) and sampled varying proportions of sentences that do not contain any hedges for our training set. We trained additional BERT models using the same hyperparameters and show our results in 3.2.

StratifiedBERT (1-to-X) is a BERT model trained on stratified data as mentioned before. We ensure that there is an equal proportion of classes (namely single and multi-expression hedges) in the training, validation, and testing set. In the columns **x-to-y** describing each model, **x** refers to the proportion of sentences that contain at least one cue and **y** refers to those that do not contain any cues. We analyse this phenomenon in further detail in section five by training and testing BERT on InfoCorpus.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
1-to-1	0.81/0.91/0.85	0.81/0.91/0.85	0.81/0.91/0.85
1-to-2	0.73/0.96/0.79	0.73/0.96/0.79	0.73/0.96/0.79
1-to-4	0.76/0.87/0.81	0.76/0.87/0.81	0.76/0.87/0.81

Table 3.2: Hedge Cue Detection results for StratifiedBERT with different ratios of data.

Finally, We identify our models' predictions and identify areas where they misclassified words. Table B.1 shows the false positives and false negatives returned by BERT trained on unstratified and stratified data.

3.2.2 Hedge Scope Resolution

We again show the average macro Precision/Recall/F1 scores for each model trained and tested on BioScope 3.3. Unlike cue detection, we are not able to utilise a similar approach for altering the distribution of data. Since we only train our models on sentences that have a scope, there is no way for us to vary the ratio of sentences that do not have a scope to those that do. However, future work for scope resolution could test these models on sentences that do not contain any scopes to verify their performance.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.93/0.99/0.96	0.93/0.99/0.96	0.93/0.99/0.96
XLNet	0.95/0.99/0.97	0.95/0.99/0.97	0.95/0.99/0.97
RoBERTa	0.94/0.98/0.96	0.94/0.98/0.96	0.94/0.98/0.96

Table 3.3: Hedge Scope Resolution results for BERT, XLNet, and RoBERTa.

3.3 Discussion

Britto and Khandelwal note a non-insignificant improvement in the F1 score when using the classification algorithms. However, during our implementation, we only saw a 0.01 increase or no increase at all when using either the *Average Token* or *First Token* method compared to the token-only method. Although these metrics do not differ in performance, we still show them for completion's sake.

3.3.1 Hedge Cue Detection

Interestingly, cue detection models reported a higher recall than precision 3.1 3.2. This suggests that models returned a high number of false positives (i.e., classifying non-cue words as cues). Furthermore, table B.1 confirms our hypothesis; hedges are context-specific (in terms of the sentence they are used in, not domain-wise). Take the word *'considered'* for example. Both models (**BERT** and **StratifiedBERT (1-to-4)**) have misclassified this word as a cue. This word sometimes appears as a hedge in BioScope and WikiWeasel, showing that *'could'* may not be domain-dependent in this instance. However, there are cases where this word is not a hedge. For example, the sentence '*We also could not identify the 5-bp TSDs and TIRs characteristic of the Transib superfamily*' contains the same word. This is not an instance of hedging, as the semantics of this word lean more towards the *capabilities* of the subject rather than the *possibilities*.

There were more examples like the above. We see that most false positive words for **BERT** and **StratifiedBERT (1-to-4)** can be used as hedges (e.g. *'potential'*, *'argue'*, etc). Therefore, this reinforces the idea that cue detection is challenging due to the contextual ambiguities of words. Even state-of-the-art transformer models struggle to classify non-cue words.

We note that we can vary the training dataset size while maintaining a consistent F1 score 3.2. In terms of the models' training times, we saw that **StratifiedBERT (1-to-2)** took around half as long per epoch than **StratifiedBERT (1-to-4**) (four minutes vs ten minutes per epoch), even though their performances were very similar. This shows us that we can efficiently reduce the size of the data without significantly harming the overall performance. We give evidence for this claim in section five.

A potential issue with reporting the average macro F1 score would be the smoothing of label scores (for cue detection) with that of either the non-cue or the padding. Since all our models perform extremely well for non-cue and padding (> 0.99 F1 score respectively), this would affect our reported average macro F1 score. This obscures the fact that the model did not learn the given inputs at all and managed to classify half the inputs correctly (when it did not).

We can treat hedge cue detection as an IR (Information Retrieval) task. Therefore, it seems appropriate to only consider labels that matter (i.e., those that we want to detect). To this end, a better evaluation metric would be to take the harmonic mean 3.1 of the P/R/F1 scores for the cue labels (single-cue and multi-cue). This takes into account the extreme values that the average would not. Therefore, we would be able to better detect whether there were any unusual outcomes in the relevant labels. Unfortunately, we were not able to use this metric in practice as some previous model files were corrupted.

$$H(x_1,\ldots,x_n) = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} \tag{3.1}$$

3.3.2 Hedge Scope Resolution

We also confirm that our models achieved similar performance to the state-of-the-art models for scope resolution. Most hedge cues in BioScope are either verbs (suggest, appear) and auxiliaries (may, might, can) 3.4.

Stemmed cue	**Count**
suggest	72
may	67
might	40
appear	38
possibl	36
like	36
whether	28
could	28
or	26
would	21

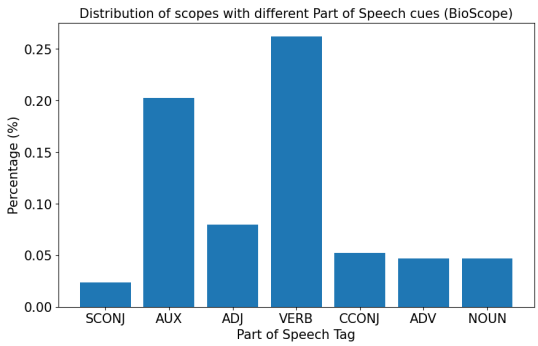
Table 3.4: Stemmed Cues and Counts in BioScope

Furthermore, we show the distribution of different parts of speech tags (e.g. `VERB`, `ADJECTIVE`, `ADVERB`, etc) in BioScope 3.1a. We see that the `VERB` and `AUX` (auxiliary - e.g. 'may' tags outnumber all other tags by a significant margin, with the `ADJ` (adjective - e.g. 'unclear') tags coming in at third. From this, we can glean that hedges are mostly auxiliary words, verbs, and sometimes adjectives.

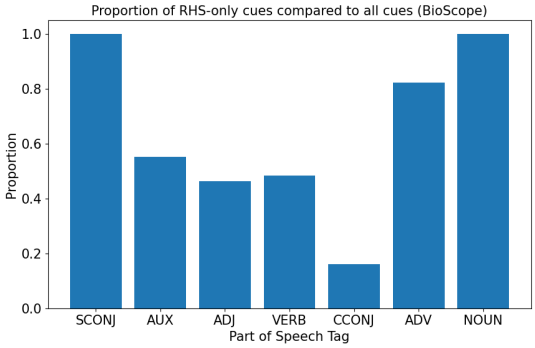
Moreover, we also plot the distribution of parts of speech tags that only have an RHS in 3.1b. We can see the all cues with `SCONJ` (subordinating conjunction, e.g. 'whether'), `ADV`, or `NOUN` tags only exhibit RHSs (Right-Hand Scope). Therefore, when we encounter cues with these part of speech tags, there is a high possibility that they only have an RHS.

On the other hand, we can see that around 50% of tags that are `AUX` or `VERB` exhibit only an RHS. This behaviour is due to the use of raising verbs or the passive voice. When we encounter these linguistic devices, we tend to incorporate the subject in the scope as well. Therefore, cues with these tendencies exhibit both an LHS and RHS. We expand on this phenomenon in section four.

Finally, we can see that the majority of `CCONJ` (coordinating conjugation, *'or'* - all other `CCONJ` words are not used in hedging) cues mostly only had LHSs, or LHS and RHS, but never only an RHS. We expected these types to show this behaviour. These conjunctions connect two sub-clauses or multiple words. Thus when we encounter a hedge with this tag, the author would be uncertain about all words that are encompassed in its scope.



(a) Distribution of scopes with respect to the Part of Speech tags of cues.



(b) Distribution of cues with Part of Speech tags that only have an RHS (BioScope).

Figure 3.1: Distribution of LHS and RHS for cues with different parts of speech tags (BioScope)

We also showcase two types of errors where the prediction differs from the ground truth:

Difference 1: Limitations of multiple scopes in a sentence

Prediction: nevertheless, the { **apparent** lack of rag2 - ... suggests that rag2 was introduced in a separate event in jawless vertebrates } .

True: nevertheless, the { **apparent** lack of rag2 - ... } suggests that rag2 was introduced in a separate event in jawless vertebrates.

In this example, the model has 'mistakenly' labelled the clause *'suggests that...vertebrates'* to be included in the scope. However, this is an unfortunate byproduct of our pre-processing stage for scope resolution. Since a sentence can have multiple cue words (and therefore, multiple scopes), we had to separate them such that there were duplicate sentences with different cues and scopes. Although the model's predictions are technically correct, we have had to classify this as an incorrect prediction as we are only concerned with the scope of the bolded cue.

A better evaluation metric would be to combine the individual scope annotations for each sentence to allow for the existence of multiple cues. However, since there could be multiple cues, we would not be able to map each cue word to its scope, thus reducing the readability of the results.

Difference 2: Failure to end at the delimiter

Prediction: ... { dl and ser have been **proposed** ... (} ...

True: ... { dl and ser have been **proposed** ... } (...

This is a simple error - the model has failed to acknowledge the delimiter (open bracket token '(') and extended the span past it.

Upon further inspection, we found that all errors fall into these two categories. From this, we can assume that transformer-based models are even better at scope resolution than the results shown in the earlier table. This is because the models will label all existing scopes of hedge cues in a sentence. Although we have labelled these incidents as 'errors' when reporting our scores, we must look at the overall picture, which is to find all scopes of hedge cues given a sentence.

Moreover, we believe that scope resolution as a task is trivial for these models. This is because many of these scopes are systematic and do not have ambiguous rules like hedge cues. For example, if the model sees a 'subordinating conjunction' (e.g. 'whether'), then the only scope possible would be from the cue word to its right until the end of the clause (i.e., it only produces an RHS according to 3.1. Transformers can learn complex linguistic phenomena due to their ability to attend to different parts of the input sentence. Therefore, they perform extremely well on scope resolution without severe faults.

3.4 Conclusion

We have successfully reproduced previous work on hedge cue detection and scope resolution. We built additional modules such as pre-processing the BioScope XML file and finding solutions to the misaligned scopes due to BPE. This was a difficult task, as the XML file was in a tricky format, and it required different iterations of pre-processing algorithms. Furthermore, we used class weights based on their frequency to train our models. We found that the two word-level classification algorithms did not provide a tangible difference during the inference period for both cue detection and scope resolution.

We also experimented with varying the input data for hedge cue detection. We increased the proportion of sentences that did not contain any hedges. The resultant models returned a similar F1 score, with **StratifiedBERT (1-to-1)** performing the best at 0.85 F1 score. We conducted this small experiment to examine the upper bound of these models. However, this would not be representative of a realistic scenario as we would not know the distribution of hedges beforehand.

For scope resolution, our models performed also performed similarly to previous work. We showed the distributions of LHS and RHS of hedge cues with various parts of speech tags. This suggested that around half of the cues were either auxiliaries or verbs and that most of them only had an RHS. Since these scopes are quite systematic, we can attribute the high performance of these models to the relatively simple task of annotating scopes.

Finally, we tentatively state that the average macro F1 score is not the best metric to evaluate our models. Since the models achieve near-perfect F1 scores for non-cues and padding, this skews the actual performance on single and multi-word cues. Since we are only concerned with labelling the cues, a potentially better score would be to take the harmonic mean of single and multi-cue F1 scores.

Chapter 4

InfoCorpus

InfoCorpus aims to combine the label types of WikiWeasel and BioScope. WikiWeasel contains information about the types of uncertainty, such as *Hypothetical Doxastic* and *Modal Probable*, while BioScope contains hedge scopes. Therefore, our dataset allows training and testing models for three tasks, the combination of which is novel (hedge cue detection, hedge type detection, and hedge scope resolution). We first introduce the corpus. We describe the annotation guidelines for cues and scopes in depth. Then, we analyse the annotations, examining some examples where annotations differed. Next, we take a step back and look at the bigger picture. We analyse whether hedge cues are domain-specific by comparing InfoCorpus to BioScope. Finally, we compare the two available corpora (BioScope, WikiWeasel) to InfoCorpus and investigate whether uncertainty types are used at different parts of a scientific paper.

4.1 Overview

InfoCorpus contains 1124 sentences from five papers in the Computational Linguistics domain. There are a total of 146 instances of hedges in 106 sentences. This results in roughly 10% of sentences in InfoCorpus containing a hedge cue and is around half that of BioScope. Even with the smaller proportion of sentences with cues, we still show that our models produce slightly lower if not similar performances for cue detection and scope resolution. The names and IDs of these papers are as follows:

We mainly chose these papers due to their similarities in domain. Not only are they Informatics papers, but they are also from the same sub-domain (Computational Linguistics). This ensures that if there are any domain-specific languages, they will be present in InfoCorpus.

We found the annotation process to be quite complex when building InfoCorpus. We describe the annotation pipeline in C.1. For example, we did not have access to monetary incentives for non-university-affiliated personnel. Therefore, we found it difficult to recruit human annotators. Furthermore, we only had access to two annotations. Although this weakens the annotations' reliability, we nevertheless achieved a high IAA (Inter-Annotator Agreement) score. Thus, we can ensure their correctness, but

Paper	ID
Can Language Models Be Tricked by Language Illusions? Easier with Syntax, Harder with Semantics [42]	1
ArchBERT: Bi-Modal Understanding of Neural Architectures and Natural Languages [1]	2
Predicting Evoked Emotions in Conversations [2]	3
BatchEval: Towards Human-like Text Evaluation [41]	4
Language Model as an Annotator: Unsupervised Context-aware Quality Phrase Generation [43]	5

Table 4.1: Paper ID Table

future work should be done to verify them. Moreover, we found that rewriting existing annotation guidelines to fit our project was challenging. This pipeline required us to build numerous scripts to filter and aggregate annotations, calculate the IAA scores and process for further annotations (e.g. cue annotations → scope annotations). Finally, building the InfoCorpus XML file itself took many iterations due to complexities with aligning the scope and cue annotations for each sentence.

4.2 Annotation Methodology

We tried to imitate the style of BioScope and WikiWeasel as closely as possible. For example, we have removed all citations where possible. If not, we removed the mentioned year. Compared to BioScope and WikiWeasel, Computational scientific papers also include Mathematical notations that the former data sets do not contain. Therefore, we have converted **in-line** equations and formulae into LaTeXnotation following Peng et al [30]. For example, the in-line formula $\{s \in \mathcal{S}\}$ would look like 4.2 in the corpus. This took a rather long time as it involved manually converting existing equations to the LaTeXformat. Fortunately, it appears that models have successfully accepted them as inputs as seen in section five.

```
\{ s \in \mathcal{S} \}
```

We employ min-max annotation guidelines laid out by Vincze et al [38]. We approach cue annotation with a minimalist strategy - i.e., we mark the smallest possible unit as a hedge cue. On the other hand, we perform scope annotation to maximise the scope size - i.e., we try to encompass the most number of words within the scope. We used Prodigy [10], an annotation software, to perform both cue and scope annotations.

4.2.1 Cue

We annotate a hedge cue to have the smallest number of words possible and mark it with the tag `cue`. Moreover, we add the uncertainty type to the `cue` tag. For epistemic uncertainty, we tag the hedge word as `e_cue`, and for hypothetical, `h_cue`. The proper hedge cue annotation for the sentence below would be the following:

It e_cue<**might**> rain.

In some cases where the hedge cue phrase is partitioned across the sentence, we label each part of the phrase as a separate cue. For example, we label the individual components of ***either...or*** as a cue.

*This could e_cue<**either**> mean that LMs strictly abide by linguistic rules to compose the language literally e_cue<**or**> that LMs have trouble understanding this complicated set of sentences overall.*

However, a hedge cue may also be used in other contexts where it does not signify uncertainty. Take this sentence:

In contrast, if the iORF is actually two adjacent genes, then upstream and downstream residues of the stop codon will appear separately in many alignments. (2)

Although we may have tagged the word *appear* in (2) as a hedge cue previously, this word is synonymous with *be seen* for example. Therefore, we must be careful when annotating hedge cues as they may or may not show uncertainty in their contexts.

Hedge cues can also be a multi-word expression as mentioned before. For example, the individual words that make up the cue in the sentence below (**to**, **the**, **best**, **of**, **our**, **knowledge**) do not convey uncertainty when used on their own. However, when joined together, they exhibit a unique definition that only this combination of words can. We label this multi-word expression as an epistemic cue.

h_cue<**To the best of our knowledge**>, ArchBERT is the first solution for joint learning of architecture-language modalities.

We show the process of annotating cues as either epistemic or hypothetical. There are four stages:

1. Normalise sentence - turn nominalisations into their respective forms (e.g. Preparation might be needed for this exam → we might need to prepare for this exam).
2. Remove cue words - (e.g. We will prepare for this exam.)
3. Format the sentence to fit the test graph 4.1 - (e.g. We **might** need to prepare for this exam, but we will not prepare for this exam.)
4. Decide whether this is true or not - (e.g. since the subject knows that they are not certain about the statement 'we will prepare for this exam', it is false that they are not going to prepare for the exam.
5. Thus, we can label the cue **might** as an epistemic uncertainty.

We now show an instance of labelling a hypothetical uncertainty. Compared to epistemic cues, this uncertainty type covers a broader range of modalities. We previously showed uncertainty types in 2.1 and we expand on them below:

- Investigative (e.g. **speculate**, **investigate**)
- Condition (e.g. **if**, **unless**)
- Doxastic (beliefs and hypotheses e.g. **assume**, **think**)

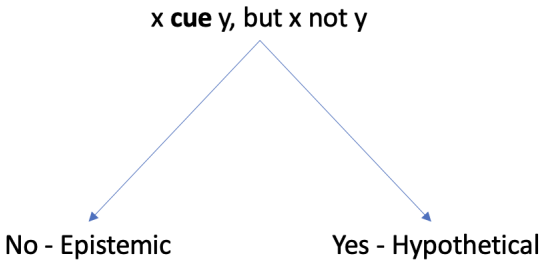


Figure 4.1: Uncertainty test graph. If the logical answer to 'x **cue** y, but x not y' is true, then the cue is Hypothetical. Otherwise, it is epistemic. This graph is taken from [3], but its usage is rather unclear. Future work should aim to describe it in more detail and make it more accessible.

- Dynamic (duties and desires e.g. **should**, **want**)

We go through the same steps as before for the sentence: 'We discuss whether LSTMs solve the vanishing gradient problem.'

1. Normalisation: N/A - there is nothing to normalize.
2. Cue removal: LSTMs solve the vanishing gradient problem.
3. Sentence formatting: We discuss **whether** LSTMs solve the vanishing gradient problem, but we are sure that LSTMs do not solve the vanishing gradient problem.
4. Use graph: Currently, the subject believes that the statement `x` **cue** `y` is true regardless of the ground truth in their world. Therefore, the cue word **whether** is a type of hypothetical doxastic uncertainty.

In practice, we did not need to go through these steps every time we suspected a word could be a hedge. For example, words such as *'may'* or *'whether'* are always used to signify uncertainty (epistemic and hypothetical respectively) in a sentence. Therefore, we labelled them as hedges whenever we encountered these words. The test proved to be more useful for ambiguous cases such as *'could'*, *'appear'*, and *'can'*.

4.2.2 Scope

{ *The chances of this happening is very* ***likely*** }

We label scopes by inserting scope tags ({ and }) at the beginning and end of the scope before any delimiters. Delimiters are punctuations or sequences of characters that signify the end of a clause (e.g. '?', ',', '.'). Furthermore, we include the hedge cue within the scope annotations. This assumes that cues and scopes are always contiguous. In the most basic scenarios, cues appear within the scopes when they directly have a subordinate clause attached to them. In more complex linguistic structures, cues might affect the entirety of the sentence, as is the case for sentential adverbs and adjectives. An example of a sentential adverb is given above. Finally, our assumption would not hold if cues exhibit long-range dependency behaviours to clauses or words, where there is a considerable distance between the cue and the clause that it affects. However, we

did not find these examples within the BioScope and InfoCorpus corpora.

Some sentences may have ambiguous semantics, such as the one below.

Mild viral e_cue<***or***> *reactive airways disease is detected.*

We could label the scope to be { *viral or reactive* }, or { *mild viral or reactive* }. Following the max strategy, we include as much of the clause as possible. Thus, we end up with this annotation. Note that the end tag (}) comes before the delimiter (full stop).

{ *Mild viral* e_cue<***or***> *reactive airways disease is detected* } .

Hedge cues with the same part of speech tags often display similar behaviours in their scopes. For example, sentential and non-sentential adjectives exhibit different scopes. Consider the following two examples:

The patient showed a { e_cue<***possible***> *case of the flu* } .

{ *The flu, however, is* e_cue<***possible***> } .

In the first sentence, the adjective **possible** is used to describe the noun phrase *case of the flue*, which results in its scope. However, the adjective encompasses the entire second sentence, and we label the full sentence as the scope.

Furthermore, when we encounter hedge cues that are verbs, adjectives, and adverbs, we label the clause to the right of the cues. Following the max strategy, we also include any adjuncts. Adjuncts are words which do not change the sentence's semantics if removed. Take this for example:

My family seemed to have had gone to the store yesterday.

In this sentence, the word *'yesterday'* is an adjunct - the sentence is complete even without the word. Thus, we would annotate it like so, where we include the word *'yesterday'* within the scope of **seemed**:

{ *My family* e_cue<***seemed***> *to have had gone to the store yesterday* } .

It should be noted that we have labelled *'My family'* into the scope. This contradicts the rule above where we label everything to the right of a verb to be in its scope. There is an exception to this rule, where we must take special care for *raising verbs*. These are verbs which *raise* the subject out of its current subordinate clause to the main clause. Some examples of raising verbs are *'seems'*, *'appears'*, *'expected'*, etc. Taking the earlier sentence:

It seems that my family has gone to the store yesterday (2)

we see that *'seems'* is a raising verb. Thus, we can raise the subject (*'my family'*) of the subordinate clause (*that my family has gone to the store yesterday*) to the main clause. We end up with the following:

My family seems to have gone to the store yesterday (3)

If we follow the guidelines, our scope would include *'my family'* in sentence 2, but not in sentence 3, which is contradictory. Therefore, when we encounter a raising verb, we must also include the subject within the scope. We would annotate sentence 3 as such:

{ *My family* e_cue<***seems***> *to have gone to the store yesterday* } .

Finally, we must take care when annotating cues in a passive clause. Take the following example:

This film e_cue<**could**> be watched in cinemas or at home.

Since there is a passive verb *'be watched'*, we must label the scopes as if the sentence was in the active. I.e., we should first normalise the sentence as such before annotating the scopes. We can now perform the annotation as normal.

We { e_cue<**could**> watch this film in cinemas or at home } .

4.3 Results and Discussion

We evaluated annotations received from two annotators. Both (involving the author) were not native but fluent in English. Before the cue detection phase, we held a meeting where we explained the annotation procedure, as well as annotated some example sentences to help the annotator familiarise himself with tasks.

Due to scheduling conflicts, one of our annotators had to drop out of the scope annotation sessions. Given that this is a student project, we could not avoid certain limitations with various resources (time, technical, etc). Therefore, we use the preliminary annotations from a single annotator (the author) for the scope labels. Although this lacks verification from a second pair of eyes, we show that scopes are more systematic than cues in the discussion. Thus, we proceed with these annotations and leave future work to compare and verify our existing annotations.

4.3.1 Cue annotations

During the annotation phase, we raised an important question. *Do the writing styles of scientific papers vary across domains?* Earlier, we suggested that certain words could signal uncertainty when used in some domains, but not others (e.g. *likely* in a Biomedical vs Mathematical context). However, this may not completely be the case. Upon inspecting the cues found in InfoCorpus 2.1, we can see that most cues here are also present in BioScope 3.4, such as *'likely'*. Therefore, hedge cues seem to not be as domain-dependent as we had previously thought.

An alternate hypothesis could be that hedges are instead dependent on the author. Authors may have different writing styles - they could concatenate brief sentences together, they could chain multiple hedges, or their writing could be extremely terse. This would make for an interesting investigation, but this is out of the scope of our project. We continue the paper under our first hypothesis and leave future work to investigate whether this claim.

We achieved a Cohen's Kappa value of 0.92 for cue annotation. This is a high score that gives us confidence that the two annotators mostly annotated the same cues. As a result, we annotated 146 hedge cues correctly in 101 unique sentences. Below, we examine and analyse different types of mismatched annotations.

Auxiliaries	Verbs	Adjectives/Adverbs	Conjunctions
May (E)	Suggest (E)	Tentatively (E)	or (E)
Might (E)	Indicate that (E)	Likely (E)	either..or (E)
Can (E)	Seem (E)	Possible (E)	whether (H)
Could (E)	Speculate (H)	Potentially (E)	
Should (H)	Assume (H)		
	Hypothesise (H)		

Table 4.2: Hedge Cue Examples (InfoCorpus). H stands for hypothetical uncertainties while E stands for epistemic uncertainties.

Difference 1: Annotation Guideline Errors

Annotator A: We explored h_cue<whether> language models capture the basic contrast between acceptable and unacceptable strings

Annotator B: We h_cue<explored> whether language models capture the basic contrast between acceptable and unacceptable strings

This difference is caused by the ambiguity in the annotation guidelines. Both annotators correctly identified that this sentence contains a hypothetical uncertainty. More specifically, this is a type of hypothetical investigative uncertainty as the author investigates or explores a certain claim. In this instance, annotators should mark the verb instigating the 'exploration' as a cue unless the conjunction 'whether' follows the verb. If so, annotators should label 'whether' as a cue since it is the smallest word (i.e., compared to 'explored whether') that suggests uncertainty.

Difference 2: Cue Span Errors

Annotator A: It also e_cue<indicates that> the quality phrases derived from these two components are complementary to some extent.

Annotator B: It also e_cue<indicates> that the quality phrases derived from these two components are complementary to some extent.

Here, we see that there is an overlap in the annotations, but the annotators did not agree upon the full span. We might think that the second annotator is correct - they follow the minimum span guideline for cues which tells us we should annotate the cue words that give us the smallest span. However, we should also remember that the span should also convey uncertainty. The first expression indicates a definitive certainty synonymous with *is*. On the other hand, the second introduces uncertainty of the subsequent clause through the use of *that* synonymous with *suggests that* or *implies that*.

Difference 3: Cue Type Errors

Annotator A: It also indicates that the quality phrases derived from these two components are complementary e_cue<to some extent>.

Annotator B: It also indicates that the quality phrases derived from these two components are complementary h_cue<to some extent>.

In this instance, we see that each annotator has labelled the cue *to some extent* differently (epistemic and hypothetical respectively). We arrive at this proposition when we use

our test graph 4.1, and we see that this proposition is false. Therefore, the correct label for *to some extent* would be epistemic.

It also indicates that the quality phrases ... are complementary **to some extent**, but it also indicates that the quality phrases ... are not complementary.

Difference 4: Contextual Ambiguities.

Annotator A: An illusion effect would appear with higher perplexity/surprisal for the unacceptable condition compared to the illusion case.
Annotator B: An illusion effect would `e_cue`<appear> with higher perplexity/surprisal for the unacceptable condition compared to the illusion case.

Finally, only one annotator has labelled the word *'appear'* as a cue in this situation. Upon further inspection, we see that the author uses the verb as a synonym of *'materialising'* rather than the raising verb *'it seems'*. Therefore, it would be correct to not label this instance of *'appear'* as a hedge cue.

Overall, we found annotating some uncertainty cues much harder to detect than others. For example, we always see *'may'* and *'might'* in a hedging context. Their sole purpose as a verb is to convey a sense of uncertainty into the clause that they dictate. However, we found it much more difficult to disambiguate words like *'can'* or *'could'* as hedges. They are more context-dependent and rely on looking at the entire clause rather than just the word.

Furthermore, we also experienced difficulties in ease of use with the test battery 4.1. We found this guideline [3] along with the descriptions of BioScope and WikiWeasel. We tried to find further provenance of this work - perhaps it stems from a linguistic or philosophical background - but we could not find any meaningful resource.

We believe that the test for hypothetical and epistemic uncertainty relates to the knowledge of **x**. By answering 'no', we are suggesting that it is not possible for **y** to be true and not true at the same time. I.e., This is not a hypothetical uncertainty (where we are uncertain about our uncertainty) and instead is an epistemic uncertainty (where we are certain about our uncertainty). Although we can arrive at a conclusive answer after careful deliberation, we believe there is much work needed to further improve and clarify this decision tree if InfoCorpus is to be built upon.

4.3.2 Scope annotations

Although these annotations lack verification, we can ensure their correctness through languages and their tendencies. Certain languages lend themselves to be more right or left-branching. Branching indicates how words form to create longer clauses and sentences. When a language is right-branching, we see parse trees such as 4.2, where phrase structures cascade to the right of the tree. Since English sentences are more right-branching than left-branching [36], we could almost always annotate the scope of a hedge cue to be the phrase structure that lies to its right.

Furthermore, we recall the distribution of scopes in BioScope 3.1. We previously found that around half of the cues with `VERB` or `AUX` only had an RHS. In cases where they also

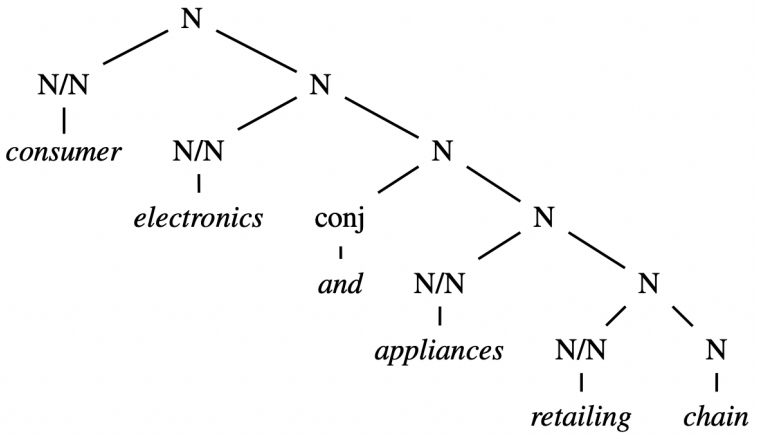
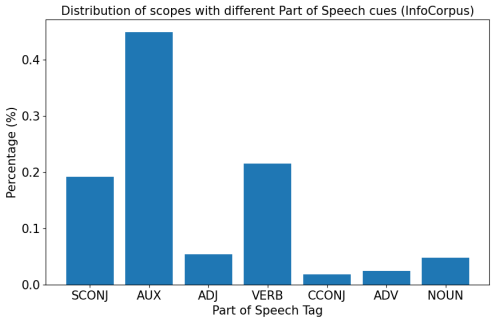
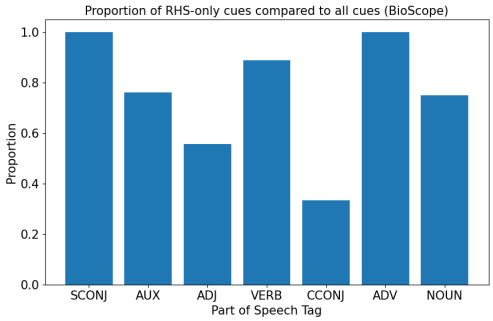


Figure 4.2: Example of a right-branching sentence [36] The tags and words themselves are not important, but rather the structure. Note how each tag expands to the right.

had an LHS, we explained that this was due to the passive voice or 'raising' verbs. For other tags such as SCONJ (subordinating conjunction - e.g. 'whether'), they universally only displayed an RHS. Thus, annotating these cases was systematic: for the LHS, we annotated the subject, while for the RHS, we annotated the clause up until a delimiter.



(a) Distribution of scopes with respect to the Part of Speech tags of cues (InfoCorpus)



(b) Distribution of cues with Part of Speech tags that only have an RHS (InfoCorpus).

Figure 4.3: Distribution of LHS and RHS for cues with different parts of speech tags (InfoCorpus).

Examining the scope distributions in InfoCorpus 4.3, we can see a similar distribution to that of BioScope 3.1. The majority of cues have an AUX, VERB or SCONJ tag. As seen in the BioScope plots 3.1b, we know that SCONJ tags almost always only have an RHS, and AUX and VERB tags have an RHS and an LHS when they are a raising verb or in the passive. This behaviour is again reflected in 4.3b. Moreover, CCONJ cues always have either an LHS or both LHS and RHS. Therefore, we simply annotate the entirety of the sub-clause to be in-scope.

Therefore, these distributions show us that certain part of speech tags exhibit an LHS, RHS, or both. For example, if we encounter a cue such as *'whether'*, we know for certain that it is only going to have an RHS, as it is of type SCONJ. Another example would be *'x **or** y'*. Here, the word *'or'* is of type CCONJ (coordinating conjunction).

Thus, we annotate the whole sub-clause. Again, these scopes are more systematic and definitive than cues, which can be somewhat ambiguous.

Although we could not source labels from other annotators for scopes, we have shown that annotating scopes is an easier task than cues.

4.4 Comparisons to BioScope and WikiWeasel

We now compare hedge cues found in InfoCorpus and those in BioScope. We see that most words are present in both corpora 2.1 4.2, but we also note one exception: **tentatively**. However, this new term is not domain-specific and perhaps was not used due to certain writing styles present in BioScope. This strengthens our hypothesis that different domains (Biomedical vs Informatics), and thus all scientific domains, may not differ in the hedge cues they use.

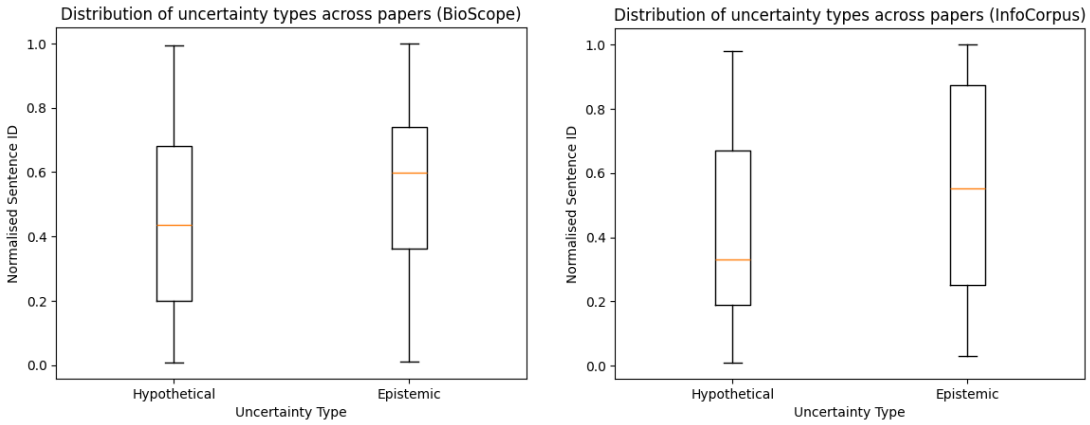
Moreover, it appears that the distributions of these cue words are similar in both corpora. Although BioScope does not contain uncertainty cue tags, we can reasonably infer that given a hedge word, it will always appear as the same uncertainty type. This is due to the test graph shown in 4.1 and the fact that there can only be one correct answer for the proposition '*x* ***cue*** *y, but x not y*'. To this end, we carried out a preliminary annotation phase, where we annotated uncertainty types into the BioScope hedges. Of course, these are not verified and future work should verify these labels.

Not only did we show that uncertainty cues appear to be domain-independent, but we can also infer that the distribution of the two uncertainty types could be domain-independent. We speculate that this phenomenon could be related to the structure of scientific papers. Earlier chapters (such as introductions) pose a hypothetical question, which is answered in later sections (such as the discussion) with epistemic uncertainty. Finally, the authors raise more hypothetical questions for future work.

Using these preliminary labels, we can partially confirm the hypothesis 4.4 (There are more hypothetical cues in the beginning and final sections of the paper). We plotted the normalised sentence ID for each uncertainty type. The equation is laid out in 4.1 where x_{ij} is the sentence line number present in paper i at line j. We divide this line number by the total number of sentences in the paper to calculate the normalised sentence ID. We see that the average ID of normalised sentences that contain hypothetical uncertainties is lower than those containing epistemic uncertainties. Thus, it appears that earlier chapters are more likely to contain hypothetical cues than epistemic cues. However, this does not confirm a higher probability of hypothetical cues in the conclusions for mentioning future work.

$$\texttt{Normalised Sentence ID} = \frac{x_{ij}}{\max_{j'}(x_{ij'})} \tag{4.1}$$

Comparing the two corpora, we see a similar trend in the distributions of cue types. However, we also see a wider spread of epistemic cues in InfoCorpus. This could either be a symptom of bad writing in InfoCorpus (e.g. using too many hedges, which leads to winding sentences) or a style that is too terse (e.g. short sentences that convey absolute



(a) BioScope (b) InfoCorpus

Figure 4.4: Distribution of uncertainty types across normalised document lengths

certainty) in BioScope. We believe that there is probably a middle ground where we can assume both sides to an extent.

We found that BioScope has a higher distribution of epistemic cues than InfoCorpus (78% and 60% respectively). This suggests that the usage of hedge cues themselves is not domain-dependent, but rather it is the type of hedge cues that are. However, although this speculation is promising, we must account for the potential errors and lack of inter-annotator agreement for the BioScope cue-type labels.

	InfoCorpus	**BioScope**	**WikiWeasel**
Hypothetical	51	150	1494
Epistemic	77	531	1771

Table 4.3: Distribution of cue types in InfoCorpus, BioScope, and WikiWeasel

We compared the distributions in InfoCorpus and BioScope to those in WikiWeasel to investigate this hypothesis. We see a similar pattern - a higher proportion of epistemic cues to hypothetical cues is present (46% vs 54%). Although these percentages are less extreme than those of InfoCorpus (40% vs 60%) and BioScope (22% vs 78%), the difference indicates that perhaps epistemic uncertainty is naturally more common in text than hypothetical uncertainty.

4.5 In conclusion

Overall, we found cue annotation to be a harder task than scope annotation. Due to the complexity of scientific papers, long and technical sentences can often be confusing when applying the test graph 4.1. Furthermore, contextual ambiguities posed major difficulties for annotators. Again, the uncertainty test graph proved to be difficult to use, and this resulted in fewer matching annotations than we would have liked. On top of this, we would have achieved a more reliable corpus had we had more annotators involved in the project.

In this chapter, we laid out our annotation process for cues and scopes. We discussed similarities and differences between InfoCorpus and other corpora. We found that the distributions of hedge cues were similar for InfoCorpus, BioScope, and WikiWeasel. We saw that there were more epistemic hedges than hypothetical ones, although BioScope recorded a more drastic distribution than the other two. However, we must also be aware that the epistemic counts are merely preliminary - future work should confirm this finding by annotating cue types in BioScope.

Next, authors seemed to use more epistemic than hypothetical cues in the latter sections of scientific papers (e.g. discussion, analysis). We did not find concrete evidence that there are domain-specific cues. This lends more credibility to section five, which tackles domain adaptation (transfer learning) of hedge cue detection and scope resolution models. We also hypothesise that the *distributions of cue types* could be domain-dependent. Authors could be encouraged to adopt varying writing styles depending on the 'recommended structure' of papers in a scientific field. This is an area that should be investigated further.

Moreover, we discovered that the distributions of scopes in InfoCorpus and BioScope were similar. `AUX` and `VERB` tags mostly only exhibited an RHS, with exceptions when `VERB` cues were used in a passive voice or they were a raising verb. We also saw that subordinating conjugates (e.g. *'whether'*) only had an RHS. An interesting question for future work could be to analyse whether the distributions of these scopes for parts of speech tags are the same for different types of writing (e.g. blogs, essays, etc).

Finally, additional work should expand InfoCorpus and ensure the annotation guideline is well-defined. Researchers should explore whether there are similarities in writing styles in each domain or whether this is entirely author-specific. This would show that we can perform domain adaptation of hedge cue detection and scope resolution models on any domain provided the writing styles are similar.

Chapter 5

In-Domain and Domain-Adapted Tasks on InfoCorpus

We now investigate whether domain adaptation is possible for cue detection and cue scope resolution. To this end, we test existing models that were trained on BioScope and test them on InfoCorpus. Furthermore, we train new models to compare the performances of in-domain and domain-adapted models. Moreover, we justify that stratifying the data set for hedge cues decreases the training time while maintaining a similar F1 score. Finally, we discuss the results from testing the models and analyse the errors.

5.1 Methodology

We perform cue and scope detection using the same methodology to reproduce previous work. That is, we train all models (BERT, StratifiedBERT, XLNet, RoBERTa) for sixty epochs with a learning rate of 3e-05, class frequency weights, and early stopping. We also introduce a novel task: hedge cue type detection. Moreover, we use BERT and StratifiedBERT to perform cue detection across domains, and all three models without stratification (BERT, XLNet, RoBERTa) to perform transfer learning for scope resolution. We perform inference on the same test set that in-domain models have been tested on. Finally, we use the three separate classification metrics (token-based, average token, first token) when aggregating the token-level labels to word-level labels.

5.2 Same Domain Results

5.2.1 Hedge Cue Detection

First, we show the metrics when training and testing BERT, XLNet, and RoBERTa on InfoCorpus 5.1. We also see these false positive and false negative tokens for **BERT** in B.3. Furthermore, we notice that **RoBERTa** has a lower precision than the other two models. We analyse this phenomenon in the discussion. We show **RoBERTa**'s false positives in B.4.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.81/0.96/0.86	0.81/0.96/0.86	0.81/0.96/0.86
XLNet	0.92/0.99/0.95	0.92/0.99/0.95	0.92/0.99/0.95
RoBERTa	0.66/1.00/0.73	0.66/1.00/0.73	0.66/1.00/0.73

Table 5.1: Same Domain Hedge Cue Detection results for BERT, XLNet, and RoBERTa.

5.2.2 Hedge Cue Type Detection

We train and test models on a novel task: hedge cue type detection. Instead of classifying whether a token is a single cue or a multi-expression cue, we classify them based on their uncertainty type (hypothetical, epistemic). We show our evaluation metrics in 5.2 and false positives in B.2.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.79/0.92/0.84	0.79/0.92/0.84	0.79/0.92/0.84
XLNet	0.75/0.98/0.83	0.75/0.98/0.83	0.75/0.98/0.83
RoBERTa	0.74/0.97/0.82	0.74/0.97/0.82	0.74/0.97/0.82

Table 5.2: Same Domain Hedge Cue Type Detection results for BERT, XLNet, and RoBERTa

5.2.3 Hedge Scope Resolution

We carry out scope resolution by training and testing three models on InfoCorpus.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.80/0.96/0.87	0.80/0.96/0.86	0.80/0.96/0.86
XLNet	0.76/0.94/0.83	0.76/0.94/0.83	0.76/0.94/0.83
RoBERTa	0.83/0.96/0.88	0.83/0.96/0.88	0.83/0.96/0.88

Table 5.3: Same Domain Hedge Scope Resolution results for BERT, XLNet, and RoBERTa.

5.3 Domain Adaptation Results

In this section, we show the results for domain adaptation of the three models in 3.1. Domain adaptation refers to training a model on one 'domain' (e.g. Biomedical) and testing it on another (e.g. Informatics). This tests the model's ability to generalise well to any domain. We perform this task to answer the question 'Can we generalise hedge tasks to more scientific domains?'. If we achieve a similar score in domain-adapted models compared to same-domain models, then we can assume that there may be a significant overlap between the hedge cues in the training and test domains. We tested saved models trained on BioScope and on the same test datasets as those used in in-domain testing.

5.3.1 Hedge Cue Detection

BERT and **StratifiedBERT (1-to-4)** returned these values 5.4 when tested on InfoCorpus. We show the false positives and false negatives of **BERT** in B.3. We could not perform domain adaptation for other models due to time and resource constraints as we had run out of compute units on Google Colab.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.69/0.80/0.73	0.70/0.80/0.74	0.70/0.80/0.74
StratifiedBERT	0.63/0.80/0.68	0.63/0.80/0.68	0.63/0.80/0.68

Table 5.4: Domain Adaptation Hedge Cue Detection results for StratifiedBERT.

5.3.2 Hedge Scope Resolution

Meanwhile, we saw much better results for **BERT** and **XLNet** in scope resolution. RoBERTa produced a peculiar result, where it failed to identify most in-scope tokens correctly. We analyse these errors in the discussion section below.

	P/R/F1	P/R/F1 (First Token)	P/R/F1 (Average Token)
BERT	0.86/0.99/0.91	0.86/0.99/0.91	0.86/0.99/0.91
XLNet	0.84/0.93/0.88	0.84/0.93/0.88	0.84/0.93/0.88
RoBERTa	0.54/0.79/0.46	0.54/0.79/0.46	0.54/0.79/0.46

Table 5.5: Domain Adaptation Hedge Scope Resolution results for BERT, XLNet, and RoBERTa.

5.4 Discussion

5.4.1 Validation for dataset stratification

Before discussing the results, we show that stratifying the dataset is a feasible method. When examining the results of **StratifiedBERT** in section three, we saw a similar F1 score while varying the distribution of the data. We could not test this claim on BioScope as we had limited resources in terms of time and compute units in Google Colaboratory. Therefore, we use this space to prove this behaviour by training **StratifiedBERT** on different distributions of the InfoCorpus dataset.

Initially, we had hoped to present ROCs (Receiver Operator Curve) of **StratifiedBERT** at various proportions to show that the model still performs well even as we lower the number of sentences without hedge cues. ROCs demonstrate the capabilities of a model with respect to varying threshold points. However, due to the sparsity of hedges in InfoCorpus (just 146 cues, 21 of which are in the test set), they did not provide a meaningful comparison between the models. We show an example plot in 5.1, which shows a near-perfect ROC as well as an AUC of 0.99. This does not clearly show the capabilities and limitations of the model and therefore is redundant to include in our project.

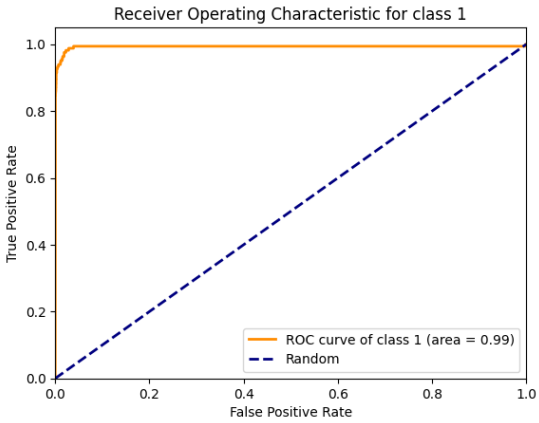


Figure 5.1: ROC of **StratifiedBERT (1-to-5** trained on InfoCorpus). Class 1 maps to single hedge cues.

To this end, we present a different plot. In 5.2, we show that the average macro F1 score of **StratifiedBERT (1-to-k)** remains consistent when we vary the dataset proportion. Barring any minor differences, we see a relatively consistent F1 score that is higher than the current state-of-the-art. Although unrealistic, this suggests that we can use less data to achieve a similar performance when performing cue detection.

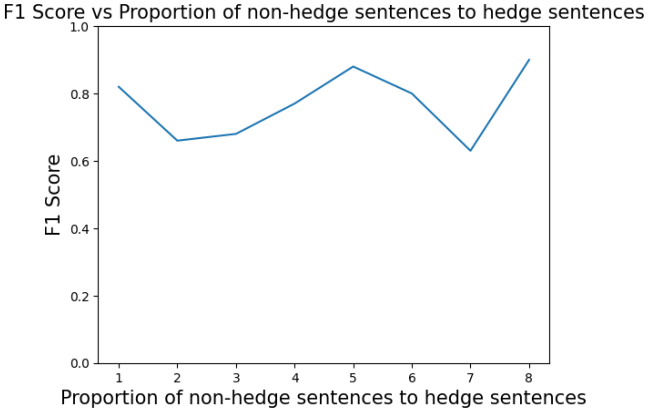


Figure 5.2: Trend of F1 score for **StratifiedBERT (1-to-k)** as we increase the proportion (k) of non-hedge cues in the training set.

5.4.2 Cue Detection

We report that domain-adapted models achieve a lower performance 5.4 than in-domain models 5.1. Although these scores do differ by around 0.10 F1 score, the domain-adapted models still correctly labelled some hedge cues. We believe that given more data, the models would perform better, but due to the constraints of annotations and time, we leave this to future work. Regardless, we can infer that domain adaptation is possible to some extent. We also hypothesise that we can extend hedge cue detection to more scientific domains. However, additional work must be done to show whether different writing styles affect the use of hedges rather than the domains themselves.

Examining the false positives and false negatives for in-domain and domain-adapted **BERT** models B.3, we can see some similarities. For example, both models incorrectly labelled *'could'* as a hedge. We also encounter some expected behaviour for **BERT** where the domain-adapted model performs worse than the in-domain model (0.73 F1 vs 0.89 F1 respectively).

We could partially attribute the lower performance of the domain adapted to the limited verification of annotations for InfoCorpus. For example, we see that the in-domain **BERT** incorrectly labelled likely cue words such as *'seems'*, *'attempts'*, and *'whether'*. Since we only took the intersection of labelled cues from the annotators, some cues were valid hedges but were not confirmed by the other annotator. Although this has resulted in a lower F1 score, we believed it was better to be certain of our annotations and achieve a lower score rather than have ambiguous labels by taking the union of both annotators' labels. This would have led to a misleading score, obfuscating the final result. As a result, with more cue annotations and a more rigorous annotation process, these domain-adapted models would have performed even better.

Curiously, we see that the in-domain **RoBERTa** reports a lower precision score compared to the other two models (0.66 vs roughly 0.87 respectively) 5.1. We examine RoBERTa's false positives B.4. In addition to expected results such as ambiguous tokens (e.g. *'likely'*, *'could'*, etc), there are some inexplicable results such as *'.'*, *'Introduction'*, and *'1'*. Similar to subsequent explanations for RoBERTa's behaviours, we believe that they stem from its tokenisation process, which differs from that of BERT and XLNet (BPE vs WordPiece, SentencePiece respectively). However, additional work should be done to verify this claim, and whether there are any other errors outside of this.

Across the board, we saw similar F1 scores for hedge cue detection. However, we must verify that this was not due to the size of InfoCorpus. Since InfoCorpus has significantly fewer hedge tokens, this results in a misleading higher evaluation score as the number of hedges in the test is low (21 hedges). Therefore, the main takeaway from this experiment is not to show that models perform 'better' on InfoCorpus than BioScope, but rather to show that hedges are domain-independent and we can generalise hedge cue detection to any scientific domain. Future work should expand on InfoCorpus as mentioned in section four to verify the models' performances are accurate.

5.4.3 Type Detection

We ventured into new areas in hedge cue detection, namely hedge-type detection. Our findings are similar to that of regular hedge cue detection - we report similar scores for all models, with the former (**BERT**), performing the best. Examining the misclassified tokens B.2, we did not notice any difference in terms of the types of errors the model made. We see that the model has misclassified words that are ambiguous hedges (e.g. *'could'*, *'should'*, etc). Future work should expand on this task. A possible extension would be to examine whether some tokens are classified more as one hedge type than the other. This would show that either the hedge type annotations are wrong, or that the test for uncertainty types 4.1 needs revising. Both scenarios tie into future work for section four, where we state that the annotation guidelines need verifying and the test battery needs a more formal explanation.

5.4.4 Scope Resolution

Across most models, we achieved a similar score of around 0.85 F1 score. This shows that scope resolution is indeed a task that can be generalised across multiple scientific domains. From this, we can infer that the scopes of these cue words in InfoCorpus are not much different to those in BioScope. Thus, we can reasonably say that scopes are domain-independent and rather cue-dependent on the cue word that they encompass.

One interesting type of result we achieved was a poorly performing RoBERTa model for domain adaptation 5.5. We saw that this model recorded a low precision of 0.54 which contributed to its poor F1 score of 0.46. Below, we identify and analyse two types of errors that we found with RoBERTa.

Error 1: Erroneous labelling of delimiter

Pred: { could distinguish acceptable sentences ... humans have no trouble dealing with } . { pad ...
Label: { could distinguish acceptable sentences ... humans have no trouble dealing with } . pad ...

The model has correctly labelled the scope *'could ... with'*. However, it has also labelled the first padding token to be in-scope. We saw multiple incidents of this error during testing. We suspect that our 'solution' of the BPE problem A.1 has caused this issue, and this raises an interesting question about the viability of certain tokenisation methods (splitting sentences into words or sub-words). It may be that certain models (like RoBERTa) are less suitable for some tasks (those that involve detection at the token-level) due to their handling of tokenisation.

Error 2: Mislabelling the <SPEC> tokens

Pred: { < SPEC > If the score is greater ...
Label: < SPEC > { If the score is greater ...

In our reproduction of previous work, we followed suit by appending a special token (<SPEC>) before the hedge cue to signify its existence. However, this seems to have interfered with RoBERTa. We believe that the reason for this could lie in the pre-processing of sentences during its training. Since RoBERTa tokenises <SPEC> into multiple token IDs, this could have introduced some problems when training on other occurrences of the tokens such as < or >.

We could use an UNK token to mitigate this problem. These token IDs are only used when the tokeniser encounters an unknown word, which would be impossible unless we manually feed the tokeniser this ID. Another approach to take would be to remove this special token altogether and investigate whether this does make a difference at all. We speculate that this would depend on whether the model can pick up on hedge cues as well as the scopes. By not including this token, each model would arguably be performing both hedge cue detection and scope resolution since it would first need to detect the hedge cue, and then its corresponding scope. Therefore, we assume that without these tokens, models would perform worse. Unfortunately, due to limited resources, we were not able to validate this claim.

5.5 Conclusion

In this chapter, we have successfully demonstrated that transformer-based models are capable of performing hedge cue detection and hedge scope resolution. Furthermore, we show that these models also can detect hedge types (a novel task to the best of our knowledge) with similar or slightly worse performance. Interestingly, we noticed that RoBERTa did not perform as well as the other two models in cue detection and scope resolution. This appears to stem from its tokenisation process, and we speculate that not all types of tokenisation are well-equipped to deal with all types of tasks. We could also attribute this to the domain-adapted model not encountering certain 'manipulated' tokens A.1 during training for hedge scope resolution.

We wanted to answer an important question in this project - 'Can we generalise hedge cue detection and scope resolution to more domains in scientific papers?'. We provided a preliminary answer in this chapter. The domain-adapted models performed at a similar level to same-domain models. Although InfoCorpus is quite a small corpus, we can tentatively answer - yes, we can.

It appears that cue detection is a harder task than scope resolution when observing the F1 scores. This is expected, as cues can be ambiguous in their semantics (e.g. *'could'*), whereas the scope of a hedge is easier to determine according to the annotations. However, there are cases where the scopes are also ambiguous.

I { **might** have seen the house } with the telescope.
I { **might** have seen the house with the telescope }.

The first sees the subject be uncertain of whether they have seen the house or not and using the telescope is a fact. In contrast, the subject is uncertain of the entire statement, that they are unsure that they saw a house, and whether they used a telescope. However, given we are concerned with hedge cues within scientific domains, we hope that the writing is clear enough to avoid ambiguities such as the above. It would still be interesting to see how these models perform when tested on scope-ambiguous sentences.

Further work in this area should focus on examining RoBERTa's failure on these two tasks when compared to its peers (BERT, XLNet). It should investigate whether some tokenisation methods lend themselves to certain tasks more than others (e.g., sentence-level predictions vs token-level predictions).

More generally, it would be interesting to carry on the discussion from the previous chapter. We have established that perhaps there are no 'domain-specific' hedges and that this entirely falls upon the author. Could we perform domain adaptation on these tasks (hedge cue/type detection and hedge scope resolution) by varying the author of the test dataset? There has been extensive work on detecting the author of an unseen text. For example, we could train BERT on a hedge-annotated corpus from an author and test it against another author's. From this, we would be able to ascertain whether hedges are style-dependent.

Chapter 6

Conclusion

6.1 Reproduction of Previous Work

In section three, we showed that it was possible to perform hedge cue detection and scope resolution on scientific papers. We analysed BioScope, which contains information about hedge cues and their respective scopes. We tested state-of-the-art transformer models such as BERT, XLNet, and RoBERTa, all of which achieved an F1 score of at least 0.78 and 0.96 for cue detection and scope resolution respectively 3.1 3.3.

Furthermore, we explored ways to vary the distribution of the dataset. We accomplished this by stratifying the number of sentences with and without any hedges to different ratios. When we applied this method to **BERT** (resulting in **StratifiedBERT (1-to-k)**), we achieved a similar score with less data 3.2. Moreover, we showed that we could maintain a relatively consistent F1 score while varying the ratio of sentences without any hedges 5.2. Again, we emphasise that this was an experiment to examine the potential of these models and that the reported scores are not reflective of a realistic scenario.

6.2 InfoCorpus

We successfully produced a dataset containing verified cue annotations and types (0.92 Cohen's Kappa Score), and preliminary scope annotations. To the best of our knowledge, this is the first corpus containing all three labels. During cue and scope resolution, we found that writing annotation guidelines was a difficult task. There should be no ambiguous instructions, and the guideline requires multiple iterations until it becomes usable. Furthermore, although the test battery 4.1 was helpful in some ways, the test became quite cumbersome to perform when disambiguating potential hedges.

We ran into several problems when aggregating the annotations. For example, we annotated some words as hedges when they were not used in a hedging context, such as *'appears'* or *'can'*. Furthermore, we disagreed on certain multi-word expressions like *'indicates that'*, the sum of which portrays uncertainty that the individual parts (*'indicates'*, *'that'*) do not.

We faced a limitation on the number of annotators available for scope annotation. However, we also showed that scopes were mostly systematic (often branching to the right in English). We found that the majority of PoS (Part of Speech) tags only exhibited an RHS in BioScope 3.1. Thus, we used this phenomenon for our annotations and showed that the distributions of scopes 4.3 were similar to those found in BioScope.

During our annotation process, we also hypothesised about the frequency of each uncertainty type in a scientific paper. We found that authors tended to use hypothetical uncertainty toward the beginning of papers and epistemic uncertainty toward the end 4.4a 4.4b. We could attribute this to the structure of the papers: we speculate (with hypothetical uncertainty) about potential results before the methodologies. Afterwards, we use epistemic uncertainty to 'soften' claims about our findings. As we now had a corpus in an unexplored scientific domain, this enabled us to test our main claim in the next section.

6.3 Domain Adaptation

Finally, we performed domain adaptation by taking reproduced work on BioScope and applying it to InfoCorpus. Domain-adapted models achieved similar, or lower, scores than in-domain models 5.1 5.4 5.3 5.5. This allowed us to answer our main claim, that hedge cues and scopes were not entirely domain-dependent, and that we could generalise them to more domains. Notably, we observed that **RoBERTa** failed to generalise well on InfoCorpus when performing scope resolution, and we showed that this could be due to the BPE tokenisation process. We analysed this in more detail, showing that the scope predictions were mostly correct, but **RoBERTa** often inserted the scope tags after delimiters or inside `<SPEC>` tokens. This implies that certain tokenisation methods might not be as well suited for some tasks as others.

6.4 Future Work

First, we should verify that **RoBERTa** can achieve a similar F1 score when compared to other models on InfoCorpus. It appears that either, there is an error in the tokenisation process, or BPE is simply not suitable for token-level tasks. Moreover, future work should strengthen the annotation guidelines and facilitate the test battery 4.1 to be more accessible. There may be either a foundation upon which this test was made or a more fitting test to use when differentiating the uncertainty types. For the corpus, additional papers must be added to InfoCorpus to increase its size, which should help with verifying the models' capabilities.

More broadly, we have observed that hedges are domain-independent in Biomedical and Informatics papers. Therefore, future work should investigate whether these behaviours are common across all scientific domains and whether we could extend this to other forms of writing. We have attributed the distribution of uncertainty types to the structure of scientific papers. Could we say the same for essays or blogs? We speculate that authors in this writing form adhere to different structures, and it would be interesting if transformer-based models could also predict hedges in these situations.

Bibliography

[1] Mohammad Akbari et al. “ArchBERT: Bi-Modal Understanding of Neural Architectures and Natural Languages”. In: *arXiv preprint arXiv:2310.17737* (2023).

[2] Enas Altarawneh et al. “Predicting Evoked Emotions in Conversations”. In: *arXiv preprint arXiv:2401.00383* (2023).

[3] MTA-Szte Research Group on Artificial Intelligence. *Hedge Type Classifications*. URL: https://rgai.inf.u-szeged.hu/file/47.

[4] Ekaba Bisong and Ekaba Bisong. “Google colaboratory”. In: *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners* (2019), pp. 59–64.

[5] Benita Kathleen Britto and Aditya Khandelwal. “Resolving the scope of speculation and negation using transformer-based architectures”. In: *arXiv preprint arXiv:2001.02885* (2020).

[6] 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)*. 2005, pp. 152–164.

[7] The SIGNILL Conference on Computational Natural Language Learning. *Previous Shared Tasks*. https://rgai.inf.u-szeged.hu/node/118. 2010.

[8] Wikipedia contributors. “Weasel word”. en. In: *Wikipedia* (Oct. 2023). URL: https://en.wikipedia.org/wiki/Weasel_word#:~:text=A%20weasel%20word%2C%20or%20anonymous,terms%20may%20be%20considered%20informal..

[9] Walter Daelemans et al. “Timbl: Tilburg memory-based learner”. In: *Tilburg University* (2004).

[10] Explosion. https://prodi.gy/. 2023.

[11] Federico Fancellu et al. “Detecting negation scope is easy, except when it isn’t”. In: *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics: volume 2, short papers*. 2017, pp. 58–63.

[12] Richárd Farkas et al. “The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text”. In: *CoNLL Shared Task*. 2010. URL: https://api.semanticscholar.org/CorpusID:549335.

[13] Christiane Fellbaum. *WordNet: An electronic lexical database*. MIT press, 1998.

[14] Viola Ganter and Michael Strube. “Finding hedges by chasing weasels: Hedge detection using Wikipedia tags and shallow linguistic features”. In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. 2009, pp. 173–176.

[15] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods". In: *Machine Learning* 110 (2021), pp. 457–506.

[16] Ken Hyland. "The Author in the Text: Hedging Scientific Writing." In: *Hong Kong papers in linguistics and language teaching* 18 (1995), pp. 33–42.

[17] Ken Hyland. "Writing without conviction? Hedging in science research articles". In: *Applied linguistics* 17.4 (1996), pp. 433–454.

[18] Feng Ji, Xipeng Qiu, and Xuan-Jing Huang. "Detecting hedge cues and their scopes with average perceptron". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*. 2010, pp. 32–39.

[19] Feng Ji, Xipeng Qiu, and Xuan-Jing Huang. "Detecting hedge cues and their scopes with average perceptron". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*. 2010, pp. 32–39.

[20] Halil Kilicoglu and S. Bergler. "A High-Precision Approach to Detecting Hedges and their Scopes". In: *CoNLL Shared Task*. 2010. URL: https://api.semanticscholar.org/CorpusID:13986201.

[21] Halil Kilicoglu and Sabine Bergler. "Recognizing speculative language in biomedical research articles: a linguistically motivated perspective". In: *BMC bioinformatics* 9 (2008), pp. 1–10.

[22] William V. Kopple and Allen Shoemaker. "Metadiscourse and the Recall of Modality Markers". In: *Visible Language* 22.2 (Spring 1988). Last updated - 2013-02-23, p. 233. URL: https://www.proquest.com/scholarly-journals/metadiscourse-recall-modality-markers/docview/1297967160/se-2.

[23] George Lakoff. "Hedges: A Study in Meaning Criteria and the Logic of Fuzzy Concepts". In: *Journal of Philosophical Logic* 2.4 (1973), pp. 458–508. ISSN: 00223611, 15730433. URL: http://www.jstor.org/stable/30226076 (visited on 10/15/2023).

[24] Marc Light, Xin Ying Qiu, and Padmini Srinivasan. "The language of bioscience: Facts, speculations, and statements in between". In: *HLT-NAACL 2004 workshop: linking biological literature, ontologies and databases*. 2004, pp. 17–24.

[25] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[26] Ben Medlock and Ted Briscoe. "Weakly supervised learning for hedge classification in scientific literature". In: *Proceedings of the 45th annual meeting of the association of computational linguistics*. 2007, pp. 992–999.

[27] Roser Morante, Vincent Van Asch, and Walter Daelemans. "Memory-based resolution of in-sentence scopes of hedge cues". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*. 2010, pp. 40–47.

[28] Hwee Tou Ng et al. "The CoNLL-2014 shared task on grammatical error correction". In: *Proceedings of the eighteenth conference on computational natural language learning: shared task*. 2014, pp. 1–14.

[29] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[30] Shuai Peng et al. "Mathbert: A pre-trained model for mathematical formula understanding". In: *arXiv preprint arXiv:2105.00377* (2021).

[31] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[32] Oxford University Press. *Definition of epistemic*. 2023.

[33] György Szarvas. "Hedge classification in biomedical texts with a weakly supervised selection of keywords". In: *Proceedings of acl-08: HLT*. 2008, pp. 281–289.

[34] Oscar Täckström et al. "Uncertainty Detection as Approximate Max-Margin Sequence Labelling". In: *CoNLL Shared Task*. 2010. URL: https://api.semanticscholar.org/CorpusID:249085.

[35] Buzhou Tang et al. "A cascade method for detecting hedges and their scope in natural language text". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*. 2010, pp. 13–17.

[36] David Vadas and James R Curran. "Adding noun phrase structure to the Penn Treebank". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 2007, pp. 240–247.

[37] Erik Velldal, Lilja Øvrelid, and Stephan Oepen. "Resolving speculation: MaxEnt cue classification and dependency-based scope rules". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*. 2010, pp. 48–55.

[38] Veronika Vincze et al. "The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes". In: *BMC bioinformatics* 9.11 (2008), pp. 1–9.

[39] Yonghui Wu et al. "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (2016).

[40] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019).

[41] Peiwen Yuan et al. "BatchEval: Towards Human-like Text Evaluation". In: *arXiv preprint arXiv:2401.00437* (2023).

[42] Yuhan Zhang, Edward Gibson, and Forrest Davis. "Can Language Models Be Tricked by Language Illusions? Easier with Syntax, Harder with Semantics". In: *arXiv preprint arXiv:2311.01386* (2023).

[43] Zhihao Zhang et al. "Language model as an Annotator: Unsupervised context-aware quality phrase generation". In: *Knowledge-Based Systems* 283 (2024), p. 111175.

Acknowledgements

I would like to thank my supervisor, Adam Lopez, who guided me through my first thesis. I could not have done it without his help.

Appendix A

Byte Pair Encoding

A.1 Example of Byte Pair Encoding

CLS	It	Might	Sl	### eet	Tomorrow	SEP
101	2009	2453	22889	15558	4826	102

Figure A.1: Tokenization for the example sentence: *It might sleet tomorrow*. Note the splitting of *Sleet* into *Sl* and *### eet*.

A.2 'Solving' the tokenisation problem

Before	After
""	" "
%	% .
%	% ,
)	) .
),	) ,
"?	" ?
'.	' .

Table A.1: Space Insertion for solving sub-word tokenization of BPE

Appendix B

False Positives and Negatives of various models

In this section, we show the false positives and negatives of models. Note that the FP (False Positive) and FN (False Negative) relate to the hedge labels and not the non-cue or padding labels.

B.1 False positives and false Negatives for non-stratified data and stratified data (BioScope) on BERT

False Positives	False Negatives
potential	be
viewed	left
or	or
not	not
either	prediction
any	likelihood
hypothesis	consider
ideal	open
can	expected
consistent	exclude
could	indication
evidence	
proposition	
predicted	
indicating	
idea	
wish	
concept	
find	
the	
estimated	
must	
considered	

(a) FP and FN for BERT (unstratified data)

False Positives (FP)	False Negatives (FN)
considered	cannot
investigates	prone
question	prediction
well	with
idea	either
indicate	or
potential	certainty
premise	estimates
assume	estimated
known	possible
predict	to
in	assign
not	interesting
inevitably	orthology *
that	
or	
no	
indication	
informative	
a	
predicted	
hypothetical	
wish	
argue	
can	
outweighs	
expected	
schematically	
concluded	
either	
indicated	
indicates	
available	
sought	
proposed	
perhaps	
little	

(b) FP and FN for StratifiedBERT (stratified data with 1-to-4 ratio)

Table B.1: False Positives and False Negatives for BERT trained on an unstratified dataset and another on a stratified dataset (BioScope). Note that * is not a misclassification, but rather an error in the pre-processing of the data. See section 3.1.1 3.1.1 for more details.

B.2 False positives and negatives of BERT for hedge cue type detection (InfoCorpus)

False Positives	False Negatives
could	conclude
should	can
likely	difficult
let	to
will	makes
implies	it
why	
investigate	
can	
that	
potential	
whether	
can	

Table B.2: FP and FN from hedge type detection for BERT

B.3 False positives and negatives for domain-adapted and in-domain BERT (InfoCorpus)

False Positives	False Negatives
support	investigate
whether	possibility
aims	investigation
mean	investigating
seems	conclude
consider	can
propose	makes
indicate	if
could	should
estimated	assume
otherwise	difficult
plausible	the
take	knowledge
proposed	assumption
appear	best
seem	our
trend	investigations
to	it
attempts	to
?	that
address	of
infer	
...	

(a) FP and FN for domain-adapted BERT. Note that the False Positives extend past *'infer'*, but we could not include the rest due to space constraints.

False Positives	False Negatives
suggests	conclude
whether	assume
these	difficult
a	it
might	investigate
either	to
possibility	can
even	makes
this	potential
may	investigating
though	
means	
hypothesize	
greater	
hope	
mean	
can	
indicate	
if	
implies	
would	
llm	
other	
could	
should	
degree	
said	
expected	
also	
will	
ensured	
phenomena	
indicates	
therefore	
it	
that	
indeed	

(b) FP and FN for in-domain BERT

Table B.3: False Positives and False Negatives for domain-adapted vs in-domain BERT (InfoCorpus)

B.4 False positives and false negatives for in-domain RoBERTa for hedge cue detection (InfoCorpus)

False Positives	False Negatives
.	
could	
Methods	
should	
implies	
Background	
1	
surprising	
that	
if	
can	
Introduction	
probability	
likely	
it	
might	
for	
is	
indicates	
...	

Table B.4: FP and FN for hedge cue detection in-domain RoBERTa (InfoCorpus). The false positives continue after '...'. It has labelled nearly every single token as a cue, resulting in a near-perfect recall.

Appendix C

InfoCorpus Pipeline

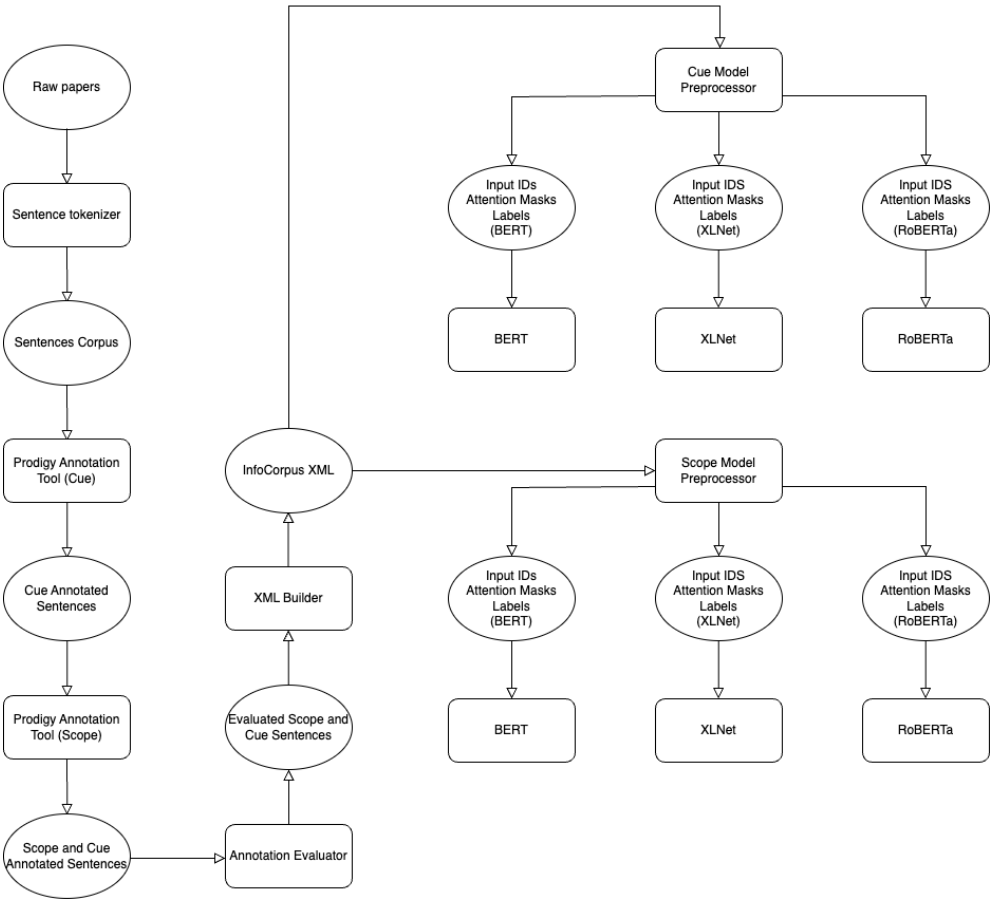


Figure C.1: InfoCorpus pipeline. This shows various data (denoted in ovals) and Python modules (denoted in rectangles). The first column relates to the pre-processing and annotation (cue and scope) of sentences. The second column evaluates the annotations and builds up the InfoCorpus XML file. Finally, the right-most columns pre-process the InfoCorpus dataset and perform hedge cue detection and hedge scope resolution using three BERT variants (BERT uncased, XLNet cased, RoBERTa cased).

Appendix D

Participation Information Sheet

Participant Information Sheet

Project title:	Develop AI-powered tools to help scientific authors write with Style
Principal investigator:	Adam Lopez
Researcher collecting data:	Daniel Kim
Funder (if applicable):	

This study was certified according to the Informatics Research Ethics Process, reference number 648491. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

The primary researcher is Daniel Kim, an undergraduate AI and CS student, and his supervisor is Dr. Adam Lopez. These are the only individuals who will have access to the data.

What is the purpose of the study?

In our study, we aim to annotate and create a new dataset, InfoCorpus, which contains information about hedge words (words which show uncertainty), hedge types (hypothetical and epistemic), and their scopes within a sentence. Using state-of-the-art machine learning models such as BERT, this will allow us to test whether hedge detection is domain-independent (i.e., not dependent on the context such as biomedical, informatics, etc) by training and testing BERT models on different domains.

Why have I been asked to take part?

You are an individual who is fluent in English, whether you are a native speaker or can speak English at a high level.

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time, up until January 15th, 2024, without giving a reason. After this point, personal data will be deleted and anonymised data will be combined such that it is

impossible to remove individual information from the analysis. Your rights will not be affected. If you wish to withdraw, contact the PI. We will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

The data being collected is comprised of individual sentences, and if present, the hedge word, its type, and its scope within it. We will collect this data through the annotation software *Prodigy*, which streamlines the process and offers an intuitive user experience for performing annotation. There are no set sessions, but rather individuals who volunteer will need to complete their share of annotations by a set date, which will be specified at a later time.

Are there any risks associated with taking part?

There are no significant risks associated with participation.

Are there any benefits associated with taking part?

There are no direct benefits associated with taking part.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be stored for a period of at least five years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team: Daniel Kim, Adan Lopez.

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted

cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who can I contact?

If you have any further questions about the study, please contact the lead researcher, Daniel Kim, H.Kim-43@sms.ed.ac.uk

If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on http://web.inf.ed.ac.uk/infweb/research/study-updates.

Alternative formats.

To request this document in an alternative format, such as large print or on coloured paper, please contact Daniel Kim, H.Kim-43@sms.ed.ac.uk

General information.

For general information about how we use your data, go to: edin.ac/privacy-research

Appendix E

Participation Consent Sheet

Participant number:_____

Participant Consent Form

Project title:	Develop AI-powered tools to help scientific authors write with Style
Principal investigator (PI):	Dr. Adam Lopez
Researcher:	Daniel Kim
PI contact details:	Email: H.Kim-43@sms.ed.ac.uk

By participating in the study you agree that:

- I have read and understood the Participant Information Sheet for the above study, that I have had the opportunity to ask questions, and that any questions I had were answered to my satisfaction.
- My participation is voluntary, and that I can withdraw at any time without giving a reason. Withdrawing will not affect any of my rights.
- I consent to my anonymised data being used in academic publications and presentations.
- I understand that my anonymised data will be stored for the duration outlined in the Participant Information Sheet.

Please tick yes or no for each of these statements.

1.	I allow my data to be used in future ethically approved research.	☐ **Yes**	☐ **No**
2.	I agree to take part in this study.	☐ **Yes**	☐ **No**

Name of person giving consent	Date dd/mm/yy	Signature

Name of person taking consent	Date dd/mm/yy	Signature

THE UNIVERSITY of EDINBURGH
informatics