COSI 140B
Professor Nianwen Xue
Group Members: Xiaoyu Lu, Yonglin Wang, Zhuoran Huang
Submission Date: May 12, 2020

Final Project Report: Temporal Relations Annotation

1 Annotation Project Summary

1.1 Project Goal and Annotation Dataset Description

Temporal relations, i.e. orders between events, are an important aspect of knowledge representation. A growing body of research in temporal relations have suggested that world knowledge of event words themselves, in addition to their surrounding words/context, also plays a crucial role in classifying the order of two events. For example, consider the following sentences where event1 and event2 both have the present simple form:

(1) The first thing I ***(event1)*** is that they ***(event2)*** writing this column.
(1a) The first thing I ***dislike*** is that they ***stop*** writing this column.
(1b) The first thing I ***ask*** is that they ***try*** writing this column.

If we replace event1 and event2 in (1) with different word pairs, it yields two opposing event orders--in (1a), event1 happens after event2, as stopping has to happen first for the speaker to then dislike it; however, in (1b), event1 happens before event2, as the asking must take place before the trying does. From these examples, we can see that we cannot simply infer the event order from the context of the event verbs, but the common knowledge of which event usually happens before the other in the real world is also crucial to our prediction.

In light of this, the goal of this annotation project is to collect as much such common knowledge on event orders, which are classified as *before, after, simultaneous,* or *vague*, as possible. Due to the constraint on the scope and time frame of our project, we have further limited our focus on annotating the event pairs in recent political news. More specifically, for this project, we have decided to annotate the order of all possible event pairs in a paragraph of a news article covering the topic of the US 2020 Election.

In this paper, we will first briefly revisit our previously established annotation schema, and we will present our the gold standard, which is an annotated and adjudicated temporal relation corpus consisting of 7 New York Times articles on 2020 elections, as well as a larger corpus that also contains potentially noisy data that extended beyond those 7 articles but were annotated by only one annotator. Then, we will present the various types of machine learning models we have trained to classify the relation between two events based on both our gold standard and possibly noisy but larger dataset. Lastly, we will discuss our overall project outcome, as well as reflecting on our experience and suggesting what could have been done to

improve this project in the future. The sections of this paper also happen to be laid out in the same order as the project timeline, which is shown in detail in Fig, 1.

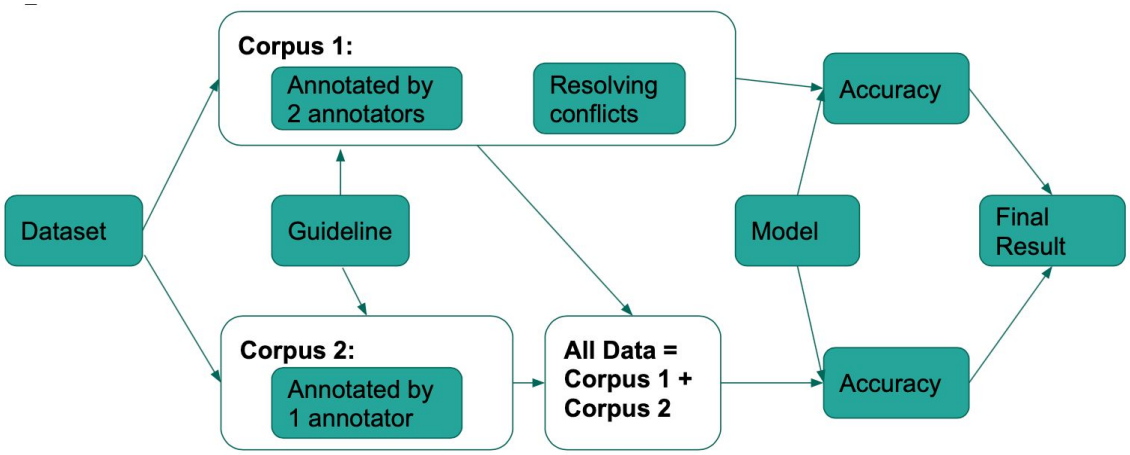

Figure 1. layout of the timeline of this project. We first gathered New York Times news articles on our selected topic, produced annotation guidelines accordingly, and had annotators annotate the data, resulting in two corpora--Corpus 1, our gold standard, and Corpus 2, our larger but noisier data. Finally, we trained our model on each of the corpora, recorded the accuracy, and presented the final result.

1.2 Annotation Schema

For the ease of execution we have chosen a simpler definition for an event that has been adopted in many papers in the previous literature (e.g. Do et al., 2012; Ning et al., 2019), which is that each event is represented by a non-auxiliary verb in the sentence. For example, in the following sentence:

> (2) As soon as the environment **becomes** stable again, the habit has **started** to **reassert** itself.

In this sentence, we have three events, represented by *becomes, started,* and *reassert* respectively. *Has* is an auxiliary verb and therefore was not considered an event.

Then we extracted all possible ordered event pairs by permuting the events in the paragraph. For example, if (2) was a paragraph on its own, then we would extract 3 ordered event pairs for the annotation process. Understandably, this way of exhausting all possibilities compromised the efficiency in annotating, an issue which we will further discuss in the last section of this paper.

Our annotators helped pick out all auxiliary verbs that cannot be filtered out by a simple script. More importantly, the annotation task was to, based on the start point of each verb in the pair, pick one of the following 4 temporal relation categories, defined as follows:

(3) Relation Types (taken from Ning et al., 2018): For an event pair (A, B), choose, **based on the startpoints** of event A and event B, one of **the four TempRel types**:

(3a) **BEFORE:** A takes place before B takes place (startpoint of A is before startpoint of B), regardless of the relation between their endpoints.
(3b) **AFTER:** A takes place after B takes place.
(3c) **SIMULTANEOUS:** A and B take place at the same time.
(3d) **VAGUE:** A might take place before, after, or at the same time as B takes place.

We have also included diagnostics tests and decision processes for our annotators to follow, some of which briefly stated below, and all of which can be found in our latest annotation guidelines.

(4) **Useful Diagnostics:**

- **Generics:** If one word in a pair is generic while the other is non-generic, we should immediately label the pair as VAGUE.
- **Possible World Analysis:** Each event should be treated as having occurred, whether or not the text implied that it had occurred. Negated events and hypotheticals are treated similarly. One assumes the event does occur, and all other events are ordered accordingly. Use context (word meaning or tense) to infer when the event would happen, if possible.
- **Causality:** If event A causes/enables/prevents B, then we say A happens before B
- **Condition:** If A is the necessary condition for B, then we say A happens before B

1.3 Adjudication and Conflict Resolution

After we collected some data annotated by more than one annotators, we began the adjudication process. For each item with conflicting annotation, we added one more blind annotation and chose the category by majority votes. If the conflict persisted, we would then discuss in the group and make the final decision.

1.4 Machine Learning Model Problem Statement

We will be training several machine learning models for our dataset, considering that it is extremely difficult to produce a rule-based system to capture the varying patterns in event ordering. With our annotated dataset split into training set and dev set, the problem statement for our machine learning model would be therefore the following: given an ordered pair of event verbs and the context, i.e. the paragraph these two verbs are extracted from, what features should be included, and which types of models may perform better at classifying the event order using the selected set of features.

2 Dataset Description

2.1 Quantitative Description

For this project, considering the restricted time frame, we expected to have nine articles annotated by two annotators; however, due to limited communication between us and the annotators, we ended up with one doubly annotated article and six singly annotated articles. By the end of the project, there were 7 annotated New York Times articles annotated by our 3 annotators. One of them was annotated by 2 annotators and the other six were annotated by one annotator. Therefore, we decided to split it into two corpora: a small but curated corpus and a larger but more noisy corpus. In order to experiment more and see the difference between the two corpora, we applied machine learning models to both the small dataset and all the data and saw the difference.

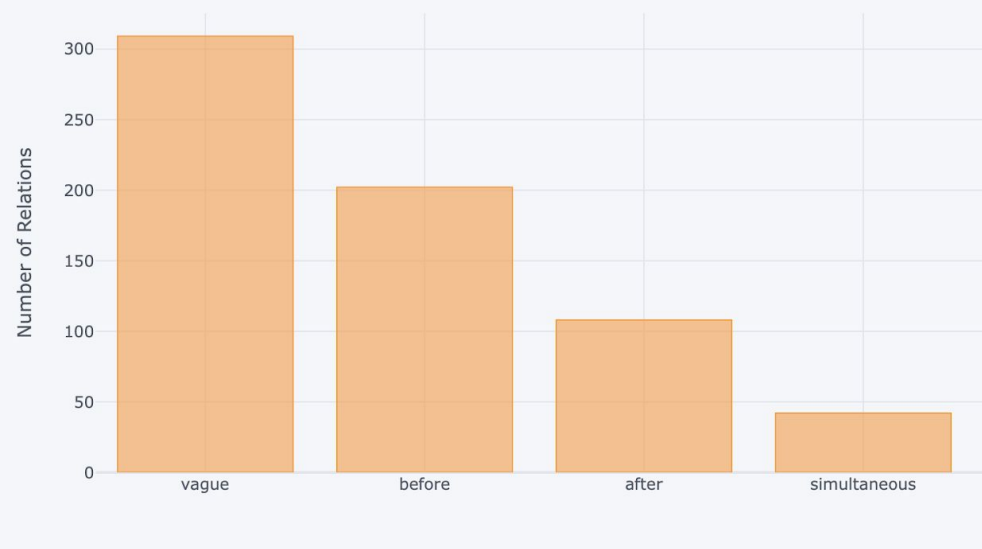

Figure 2. Distribution of the 4 labels in the Gold Standard corpus. Total number of labels = 662.

The statistics for the two datasets are as follows: for the Gold Standard which contains 662 items annotated by two annotators after adjudication, there are 309 data labelled “vague”, 202 labelled “before”, 108 labelled “after” and 43 labelled “simutaneous” (Fig. 2). For the other six singly articles, we have 2238 data labelled “vague”, 1128 labelled “beforer”, 558 labelled “after” and 543 labelled “simultaneous” (See “original” in Fig. 3 below for a visual representation).

From Tab. 2 below, we could see that the category distribution is quite imbalanced. “Vague” labels account for half of the data, which in turn makes “vague” our majority label. Therefore, for the later machine learning models evaluation, we need to note that the majority baseline is therefore 50%, and hence a good model is expected to be able to beat this baseline.

Table 1. Label count and percentage in the two corpora, Gold Standard and the larger corpus.

	vague	**before**	**after**	**simultaneous**
Gold Standard Count	309	202	108	43
Gold Standard Percentage	50%	32%	17%	7%
Larger Corpus Count	2238	1128	558	543
Larger Corpus Percentage	50%	25%	12%	12%

2.2 Dataset Quality Evaluation

Table 2. Number of Gold Corpus items receiving respective labels from the two annotators. Each cell represents the number of items given column label by Annotator 1 and row label from Annotator 2. Numbers on the diagonal therefore represent the amount of items receiving the same labels.

		annotator 1				
		before	**after**	**simultaneous**	**vague**	**total**
annotator 2	**before**	103	6	11	51	171
	after	23	41	8	39	111
	simultaneous	7	3	10	11	31
	vague	58	34	24	233	349
	total	191	84	53	334	662

For the gold standard annotated by two annotators, there are some conflicts between the annotations. As shown in Tab. 2, there are 275 out of 662 items having different annotations. Our two annotators disagree with each other, especially in the judgement of “after” and “simultaneous”. For both categories, there are only around one third data items that they agree on the judgements. For the other two categories: “before” and “simultaneous”, they agree on the labeling of two-thirds of data items which doesn’t show a high agreement either.

To evaluate the annotation of the annotated dataset, we need to calculate the inter-annotator agreement to see how much homogeneity or consensus exists in the ratings given by various judges.

Specifically, we use Cohen’s Kappa to indicate the inter-annotator agreement. For this annotated corpus, Cohen’s K is around 0.345, which lands, at best, near the fair/good range (0.4-0.750 according to Fleiss (1967)). Therefore, we could conclude that there might have been too little agreement between the annotators, which leads the annotation quality to be quite unsatisfying. One reason could be that there were a lot of subjective judgements involved in the annotation process, which led to annotators having different opinions on the same data. We will further discuss this issue in the last section.

3 Model Training

3.1 Model Description and Feature Engineering

We experimented with several models, including the generative model (Naïve Bayes), discriminative model (Logistic Regression), tree-based models (Random Forest and Gradient Boosting).

We used a total of 6 features of 3 types, namely, lexical features, syntactic features, and semantic features. More specifically, we had 3 lexical features: 1) The part-of-speech (POS) tags from each individual verb and from its neighboring two words, 2) the modal verbs between the event mention (i.e., will, would, can, could, may and might), and 3) the temporal connectives between the event mentions (e.g., before, after and since). We used one syntactic feature--the distance between the two verbs in terms of the number of tokens. For semantic features, we used two: 1) whether the two verbs have a common synonym from their synsets in WordNet (Miller, 1998) and 2) whether the input event mentions have a common derivational form derived from WordNet. We extracted these features and saved them into a .csv file.

3.2 Training Parameters

Because our task is a multiclass classification problem, so for logistic regression, we set the multi_class parameters to 'multinomial'. We also tried other parameters for Naïve Bayes and Logistic Regression, but it compromised the performance, so we set them back to default. For random forest, we used grid search to find optimal parameters. The estimators are set to 43, minimum samples split to 2, and the max depth to 500.

3.3 Experimental Settings and Training Results

The overall accuracy of the model is 0.53 for data in Gold standard corpus and 0.50 for all data annotated. Because of the high majority baseline, we also tried the resampling techniques to deal with the imbalanced data. We chose the Synthetic Minority Over-sampling Technique (SMOTE), and we applied it *only on the training data* to avoid data leakage, i.e. no information would leak from test data into the model training. We implemented this oversampling technique on the all data corpus, and the oversampling results are shown in Fig. 3 below.

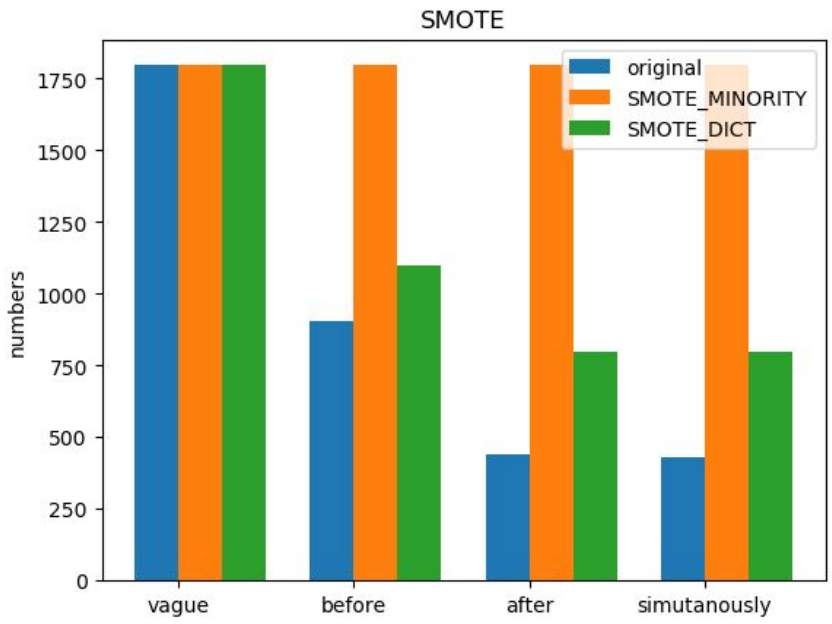

Figure 3. Numbers of data points in each category before and after two Synthetic Minority Over-sampling Techniques (SMOTE), the first one is minority and the other used dict. Minority is the technique where smaller/minority classes are resampled to majority count. Dict means that we manually set the numbers of data 'before' from 906 to 1100, 'after' from 438 to 800, and 'simultaneously' from 432 to 800.

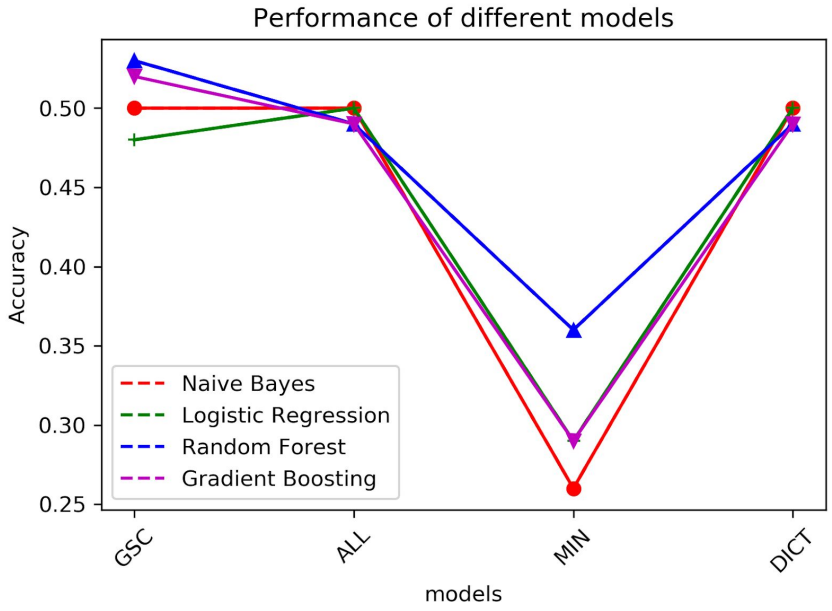

Figure 4. Performance, in terms of accuracy, of the four models used in this project on different corpus with different resampling techniques (SMOTE_MINORITY or SMOTE_DICT). GSC=Gold Standard Corpus, ALL=ALL DATA, MIN=ALL DATA(SMOTE_MINORITY), DICT=ALL DATA(SMOTE_DICT).

Table 3. Performance, in terms of accuracy, of the four models used in this project on different corpus with different resampling techniques (SMOTE_MINORITY or SMOTE_DICT).

	Gold Standard Corpus	**ALL DATA**	**ALL DATA (SMOTE_MINORITY)**	**ALL DATA (SMOTE_DICT)**
Naíve Bayes	0.50	0.50	0.26	0.50
Logistic Regression	0.48	0.50	0.29	0.50
Random Forest	**0.53**	0.49	0.36	0.49
Gradient Boosting	0.52	0.49	0.29	0.49

From the data presented in Tab. 3 and Fig. 4, we can see that we can achieve the highest accuracy of 0.53 when we train a random forest model on the Gold Standard corpus, which is slightly higher than the majority baseline of 0.5 in the Gold Standard corpus. Fig. 4 also suggests that Random Forest performs the best in three out of four different training data. Therefore, we can also stipulate that Random Forest may be a promising candidate for future model training on Temporal Relation classification tasks.

From Fig. 4, we can also observe that each model's performance lowers on when we set the resampling SMOTE parameter to Minority, where all data labels are treated as having the same count. Resampling is supposed to amplify the patterns in less represented data and give those minority patterns a higher chance of being learned. Unfortunately, this does not seem to be the case in our model training, which might hint at an insufficient amount of training data that prevented our model from acquiring any significant patterns.

Overall, we can see that our models generally performed better on the curated, Gold Standard corpus than on the larger dataset. This suggests that our adjudication process seemed to indeed make the data "cleaner" and contain a clearer pattern for the model to capture. However, it is also worth noting that even our highest model accuracy is only a tad higher than the majority baseline, which suggests that our model could be pervasively assigning a "vague" label to every item it encounters.

Table 4. The performance on each temporal relation category of the Random Forest model trained on the Gold Standard corpus (left) and all data (right). Support indicates the number of predictions for each label.

Corpus 1	precision	recall	f1	support
vague	0.82	0.54	0.65	94
before	0.40	0.55	0.46	31
after	0.11	0.40	0.17	5
simu	0	0	0	3

ALL DATA	precision	recall	f1	support
vague	0.81	0.50	0.62	719
before	0.18	0.33	0.23	120
after	0.06	0.33	0.11	24
simu	0.12	0.42	0.19	31

To further examine whether our model is indeed pervasively assigning one single label to everything, we also performed a more detailed analysis on our best performing model (Random Forest), as shown in Tab. 4. If the pervasive assignment is true, we would expect to find a large proportion of “vague” labels being assigned by our model. This is indeed the case, since out of 133 total predictions on the test set, the model assigned an overwhelming 94 “vague” labels. More unfortunately, the low recall on both datasets suggests that our model not only pervasively assigns “vague” labels, but it also fails to recognize actual “vague” items, meaning that the assignment of “vague” labels is both pervasive and random. This seems to point to the fact that the problem stems from our training data and/or feature selection, which did not provide the model with sufficient information.

Another observation from Tab. 4 is that the F1 score of each label is proportional to the distribution of that label in the training date, which means that the more represented a label is in the dataset, the better our model is at making predictions for it. This provides further evidence to our annotation schema and data collection that the dataset is very imbalanced and therefore, the schema needs to be improved and the dataset to be increased in size.

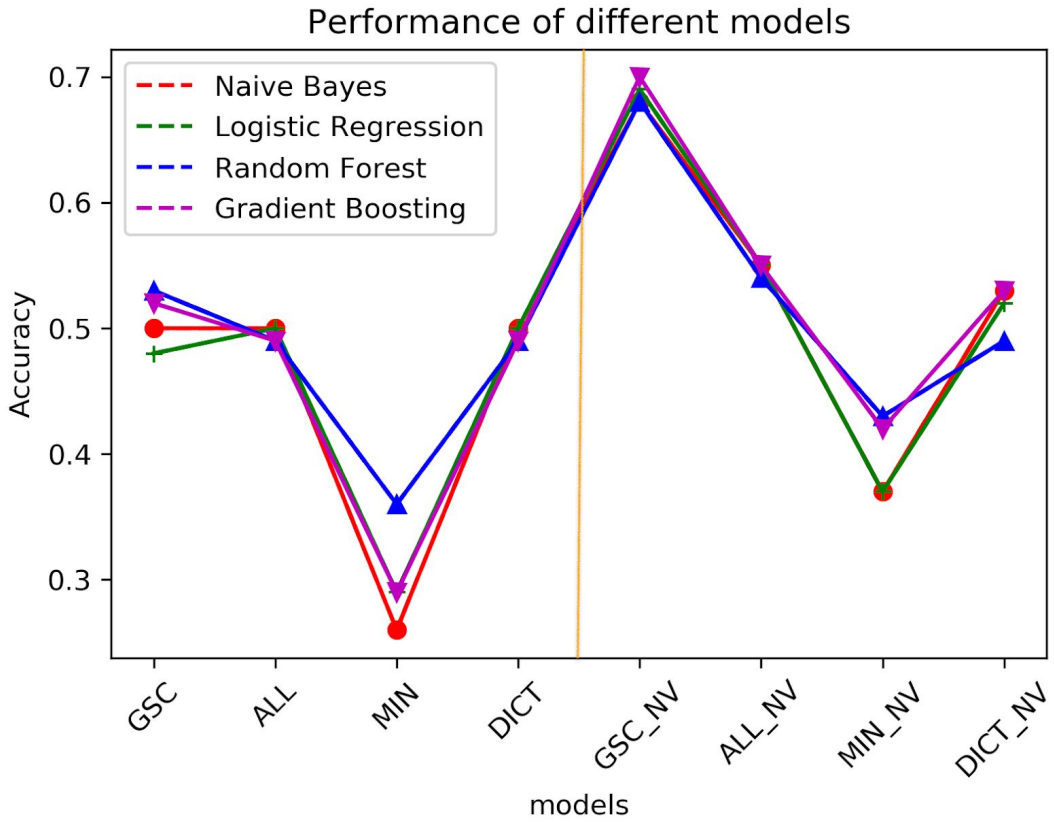

Figure 5. Performance, in terms of accuracy, of the four models used in this project on different corpus with different resampling techniques (SMOTE_MINORITY or SMOTE_DICT). GSC=Gold Standard Corpus, ALL=ALL DATA, MIN=ALL DATA(SMOTE_MINORITY), DICT=ALL DATA(SMOTE_DICT), NV=No Vague Labels.

To further understand the role that “vague” labels play in our corpus and also to determine whether there are indeed any patterns in other labels, we created a new toy dataset by excluding all the “vague” data points from the Gold Standard, while the other labels, “after”, “before”, and “simultaneous”, remain in the toy dataset. The majority label in our dataset is “before,” yielding a majority baseline of 0.57.

The results for the toy dataset are shown in Fig. 5, where we can see a clear rise in model performance. For example, the performances of four models trained on the Gold Standard corpus (see GSC_NV) are between all **0.68-0.70**, which is significantly higher than both the majority baseline of 0.57 and the results from corpora with “vague” labels. This suggests that the underlying patterns of “before”, “after” and “simultaneous” exist but have become difficult for the model to learn when the “vague” labels are also present in the corpus. Moreover, the performances on the larger all data corpus have slightly improved after removing “vague”, but are still far behind the scores of the Gold Standard corpus without “vague” labels. It could mean that our all data may not have been consistent enough for the “vague” labels to hold any sensible patterns.

Finally, we will return to the discussion of why our models had relatively low performances in general. There are several reasons for low accuracy. First, our data is imbalanced, so it is hard for our machine model to find patterns to learn. As shown in Figure 3, we have too much ‘vague’ data. For instance, some events have no relations (generic v.s. non-generic), or some might have relations but was labeled as vague because the guideline might be not thorough and clear. Second, for the event extraction, our event is not that clean because it still contains some auxiliary verbs. Additionally, due to the time constraint, we were unable to find a way to efficiently represent the verbs using vectors while not compromising the representation of other features in the input. Therefore, we in fact only used the context as features, so it might omit some essential features of temporal relations. For the feature selection process, it would be better to include the vectors of the actual words themselves, such as using word embeddings, in order to improve the model performance.

4 Conclusion and Discussion

Data play an essential role in machine learning tasks. Therefore, a clear and well-labelled dataset is the basis and key to achieving good performance. As discussed in multiple sections above, we find that the “vague” category makes up 50% of the whole corpus. The imbalanced distribution of labels in the dataset has, in turn, become a prevalent issue for our model training and analysis. Over our discussion above, we have identified some potential factors that may have led to the imbalance.

First, the data were not fully preprocessed, more specifically, although our annotators’ main task was to annotate the relation between two events, defined as two non-auxiliary verbs, our actual data items, even after manual cleaning, still contained some auxiliary verbs and some other non-verb words as events; thus, among the four labels our annotators were advised to label them as “vague.” This could have resulted in more noise in our annotated data.

Secondly, as discussed in section 3.3, we experimented to see how “vague” affects the classifiers to capture the patterns of each class. The results’ comparison of two groups, namely data with and without the “vague” label, suggests that “vague” did play an essential role in the low performance.

We have also noticed that the size of the corpus is quite small and there is a very low inter-annotator agreement. We suspect that there isn’t enough data for machine learning models to learn the patterns for classification. In order to establish a corpus that contains more consistent patterns in the future, it is highly advisable that we hire more annotators to achieve a higher amount of annotation.

Moreover, there is much ambiguity in temporal relations annotation, which makes the annotation process extremely difficult. The ambiguity lays in both the temporal relation itself and the guideline. For many cases, the relationship is not shown clearly and straightforwardly. Consequently, our annotators would each have their own distinct interpretation of the guidelines. This calls for a much more thorough and clearly defined guidelines in the future annotation projects.

To help our annotators annotate data more precisely and consistently, we need to do more research on temporal relation annotations to include as many cases that annotators might meet as possible. But it is true that it is impossible to have all the cases. For the ambiguous cases that are not defined in our guideline, we need to provide a space or label for our annotators to place those undefined data. Then we could look back to determine the judgement of such cases and modify the annotation iteratively. So our annotators only annotate the data that they feel confident with the judgement. Another way to improve that could be hiring more annotators to annotate one data item and choose the one with the majority vote. Besides that, we need to spend more time on training annotators, so that our annotators would be more familiar with the guideline and the annotation would be more consistent. Finally, we could also look at different Temporal Relation annotation techniques that have more advanced structural representation of events than ordered pairs, such as temporal dependency structure (TDS), which could minimize repetitive work and provide a clearer temporal relation between all events.

5 References

Do, Q. X., Lu, W., & Roth, D. (2012, July). Joint inference for event timeline construction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 677-687). Association for Computational Linguistics.

Fleiss, J.L. (1981). Statistical methods for rates and proportions (2nd ed.). New York: John Wiley. ISBN 978-0-471-26370-8.

Miller, G. A. (1998). WordNet: An electronic lexical database. MIT press.

Ning, Q., Wu, H., Peng, H., & Roth, D. (2018). Improving temporal relation extraction with a globally acquired statistical resource. arXiv preprint arXiv:1804.06020.

Ning, Q., Feng, Z., & Roth, D. (2019). A structured learning approach to temporal relation extraction. arXiv preprint arXiv:1906.04943.