# Detecting Temporal Relations of Events in Short Narratives
CS229 Fall 2016 Final Project Report

Delenn Chin Kevin Chen
{delenn, kchen8} @stanford.edu

**Abstract**

Event detection and temporal classification has long been a fundamental goal in NLP. Recent studies have highlighted the challenges that modern approaches to this task face, particularly when addressing both detection and classification together. Here, we use a new annotated corpus, StoryCloze, to train classifiers capable of classifying relationships spanning the length of a short narrative. Our final classifier achieves 62% test accuracy with a selection of hand-built features designed to capture lexical and syntactic features, despite sparsity in the dataset.

## Introduction

The translation of ideas expressed in natural language to a computationally usable form remains a fundamental goal NLP. Story understanding is a specific instance of such a translation, but has seen several challenges in the past due to a heavy emphasis on events and their relationships and the varying span of textual relationships. However, advances in semantic NLP techniques have made possible new approaches, regenerating interest in the task.

Temporal relation ordering is motivated by various real world applications, such as medical diagnosis and information pooling from news articles [1], [2]. As most of the data is unlabeled, there is a large interest in automated labelling of events and relationships in raw text data. We are therefore interested in using supervised methods to learn events and relationships for use in unsupervised annotations.

In this study, we aim to use a new corpus designed for story understanding and narrative structure learning to capture and learn temporal relations between common daily events. The input to our algorithm is a simple 5-sentence narrative (Fig. 1). We use multi-class logistic regression to learn common events and their relationships to output a predicted temporal relationship, {“BEFORE”, “DURING”, “AFTER”} between each pair of events.

## Related work

Previous studies have attempted to classify temporal relations with hand built semantic and syntactic features and standard machine learning classifiers and approaches, using annotated datasets [1]–[3]. These studies have reported moderate successes with simple classifiers, have suggested that increasingly sophisticated methods have failed to perform better on this classification task. The possibility that simpler methods may be more effective at classifying temporal relations serves as a clear indicator for a first-attempt classifier for this task, but also suggests that there may be underlying relationships that have yet to be captured.

Contrastingly, recent studies have focused on the extraction and classification of temporal relationships from raw datasets [4], [5]. These studies have focused on the use of split classifiers, in which specialized classifiers are used for each task (event and relation extraction) separately, to achieve higher performance. The use of split classifiers to both identify events and classify their relations presents

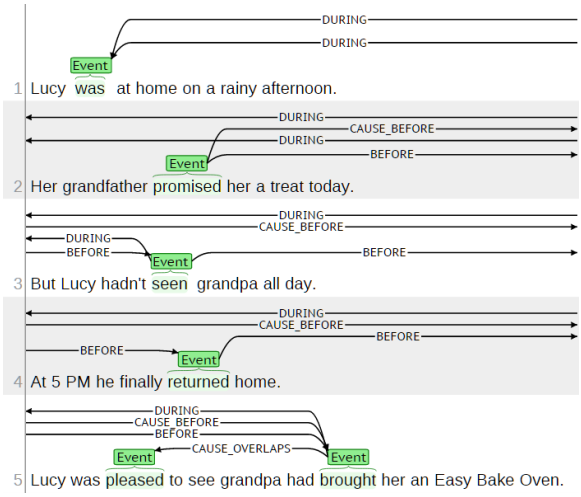


Figure 1. Sample annotated short narrative from StoryCloze corpus. Events are labelled in green, relationships shown with connecting arrows.

a dilemma – any errors in event extraction will impact the performance of the subsequent event labelling. Indeed, these studies have had difficulty achieving high accuracy on relation labeling (> 50%).

The features used in the specialized classifiers are analogous to those found in studies with annotated datasets. It is noteworthy that both studies train each classifier on separate datasets, which allow them to leverage information across different corpora during the temporal relation classification task. As temporal relations are highly dependent on event context and writing style, the use of multiple datasets may be crucial to a robust classifier.

## Dataset and Features

*Dataset*

Mostafazadeh et al. recently generated a new corpus, StoryCloze, designed for use with story understanding work [6]. The corpus is a collection of isolated 5 sentence stories, in which events in later sentences have some dependency (temporal, causative, etc.) on previously mentioned events. The corpus is released with separate annotations indicating the true set of events and relationships, allowing the use of raw sentences and annotations as algorithmic input. It is noteworthy that the corpus is limited at a total of 2,412 event relationships, spread over ~300 stories.

While the StoryCloze corpus annotations use causal temporal relationships, we focus the classification problem here by considering only temporal relationships between events. We are interested in classifying each pair of events $(e_1, e_2)$ in each story as one of {“BEFORE”, “DURING”, “OVERLAPS”}, indicating the temporal placement of event $e_1$ with respect to event $e_2$.

Notice that most events have a before-and-after relationship, where the second event in a consecutive pair occurs after the first event (Fig. 1). In the provided example, the two exceptions we see are the first and last consecutive event pairs. In the first pair, “named” and “was” refer to Bill’s state and existence, and can be said to occur simultaneously, while in the last pair, “passed away” occurs after “resuscitation.” On a broader scale, this example illustrates the difficulty involved in story understanding scope. In contrast with other well-defined NLP tasks, such as sentiment classification, temporal relations are often influenced by general syntactic structure, necessitating features capturing structure in addition to token presence

*Feature Selection*

As our study focuses on the relationships between events, we first turned our attention to the labelled events in each story. For each labelled event, we consider both the word used and the word lemma. We also draw upon synsets (synonyms) derived from the WordNet corpus to consider the relationships associated with related words.

To capture the pair-wise event relationships, we used three features specific to event pairs. First, we use tense comparisons between two events $e_1$ and $e_2$. Intuitively, this is designed to address cases when the tenses of the two events do not match, such as when one event is referred to in the past tense and the other in the present tense. We also consider event ordering within the document. While some events may be biased towards certain orderings,

alternative sentence structures resulting in swapped orderings may indicate different temporal progressions.

We noticed that syntactic structure often influences the temporal relationships between events in narratives. We therefore added two features aimed at capturing general sentence structure. Taking a cue from Chambers' previous study, we used uni-, bi-, and trigram part of speech tags to capture any syntactic similarity that occurs [5]. Constrained by limited corpus size, we also used token distance between events as a simple structure metric.

**Methods**

We first implemented a baseline algorithm using standard multi-class Naive Bayes with our selected features. We used the likelihood function

$$\mathcal{L}(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}, \phi_{j|y=2}) = \prod_{j=1}^{m} \max_y p\left(\phi_{(j|y)}\left(x^{(i)}\right)\right) p(y)$$

to assign the maximize likelihood class to each example. More specifically, for each event pair, we compute the conditional probability of the features of the event pair, given each class label. We then assign the highest probability class label to the event pair, and evaluate precision (p), recall (r), and F1 metrics with

$$p_{class} = \frac{\#\ correct\ in\ class}{\#\ predicted\ class}$$

$$r_{class} = \frac{\#\ in\ class\ correctly\ predicted}{\#\ in\ class}$$

$$F1_{class} = \frac{2 * p_{class} * r_{class}}{p_{class} + r_{class}}$$

$$F1_{avg} = \frac{1}{k}\Sigma_{i=1}^{k} F1_i$$

To better understand any class specific failings, we compute accuracy and recall for each class respectively, and average them for computation of the F1 metric across the entire dataset.

While Naïve Bayes is often used for sentiment classification and other well-defined NLP tasks, we found it to be a poor model choice for temporal relation classification. The Naïve Bayes model assumes feature independence, and is highly influenced by dataset bias towards any one class. Temporal ordering is often influenced by sentence structure as opposed to token presence and counts, which may break the assumed feature independence. Furthermore, the tendency for temporal linearity in story telling makes gives a strong bias towards the "BEFORE" class, making Naive Bayes a poor fit for the features and dataset selected.

We decided to experiment with a multi-class logistic regression, where the model does not assume feature independence. To reduce the impact of dataset bias towards the "BEFORE" class, we use a multinomial generalized linear model (GLM) to predict class probability given observed examples and derived features. We use the generalized multinomial regression modelled by Softmax Regression, maximizing the probability function

$$p(y = j|x;\theta) = \frac{e^{\theta_j^T x}}{\Sigma_{j=1}^{k} e^{\theta_j^T x}},$$

where $j = \{1, 2, 3\}$ corresponding to the three classes relationship classifications {"BEFORE", "DURING", "OVERLAPS"}.

Given the conditional probability formula for each class given an example *x*, we classify each example with the highest probability class after learning the maximum likelihood estimate parameters $\theta$. Using the Softmax probability function, we compute the maximum likelihood estimate of the parameters via the likelihood function

$$\ell(\theta) = \Sigma_{i=1}^{m} \log \prod_{l=1}^{k} \left( \frac{e^{\theta_l^T x^{(i)}}}{\Sigma_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \right)^{1\{y^{(i)}=l\}}$$

where k = 3, corresponding to the number of classes. We then evaluate training and testing performance of the multiclass logistic regression model similarly through precision, recall, and F1 metric as stated above.

**Experiments and Results**

All models were trained via 10-fold cross validation on the annotated Story-Cloze corpus, and the best model was then trained on the entire training set. Reported results are derived from a held-out test set, also taken from the StoryCloze corpus.

Our first experiment focused on the vanilla multiclass Naïve Bayes Classifier with event specific features and part of speech *n*-grams. The classifier achieved an overall F1 score of 50% on the test set (detailed results in Table 1).

Our subsequent experiments used the multiclass logistic regression model modelled by Softmax Regression. The baseline logistic regression model achieved an averaged F1 score of 48%, and inclusion of the features increased the averaged F1 metric to 62%.

**Discussion**

While our initial results with logistic regression suggest that Naïve Bayes is a stronger fit for the task, we rationalize that the performance loss is due to inherent bias in the dataset. The probabilities in the Naïve Bayes model are contingent on class probabilities determined by the dataset. As the majority of labelled event relations in the corpus have the "BEFORE" relation, Naïve Bayes is able to achieve marginally higher accuracy by simply predicting "BEFORE" more often.

Our initial experiments with features specific to events and pairs of events

Table 1. Logistic Regression Classifier performance with specified features and L1 regularization over 354 test relations

| | CLASS | P | R | F1 |
|---|---|---|---|---|
| | Before | 0.64 | 0.66 | 0.65 |
| | During | 0.62 | 0.58 | 0.60 |
| | Overlap | 0.48 | 0.55 | 0.55 |
| **AVG** | -- | 0.62 | 0.62 | 0.62 |

produced only minimal gains in performance (results not shown), leading us to pursue syntactic features. Despite their success in previous studies, part of speech *n*-grams provided only minimal gains (~1%) in our experiments. Examination of feature weights and counts after training revealed highly sparse features, which prompted us to use the simpler token distance metric to capture general structure (Fig. 2).

Realizing this, we ran experiments using only the top 100 and 200 most influential features (not shown). While these classifiers used only features with the largest weights, they were unable to achieve the same level of performance as the classifier with the full set of features ($F1_{avg}$ = 0.57, 0.60, respectively). These accuracies suggest that most the classifier's performance stems from the top 200 features, where the vast majority of our features contribute to ~2% of total performance (Fig. 2).

Surprisingly, token distance between event pairs alone increased averaged F1 performance by ~8%, making it the most influential feature in our classifier. As token distance broadly captures narrative structure across all five sentences, there may be an underlying relationship between event separation and temporal ordering. While this result appears to highlight the importance of syntactic features to the success of this classification task, these gains seem intuitively disproportionate to the simplicity of token distance. We posit that the value of token distance feature may be potentially overestimated in this study because of the

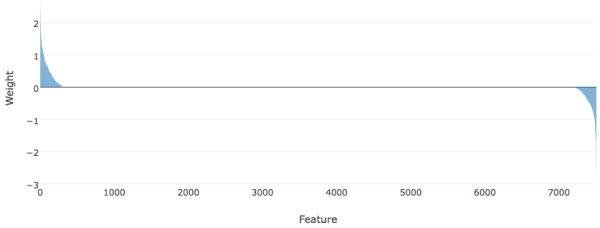


Figure 2. Plot of sorted feature weights. Most features ( > 75%) have nearly no weight.

limited size of the annotated StoryCloze corpus.

We found that the primary challenges we faced in these experiments were feature sparsity resulting from limited data and feature engineering for sentence structure (Fig. 2). With only 300 stories and 2,412 labelled relationships in the corpus, there was not enough information for simple classifiers to perform exceptionally well (machine learning typically shines with > 10,000 examples). The variety and complexity in the English language makes it difficult to succeed in a machine learning task with such a small data set.

A possible approach to address the lack of information available in StoryCloze is to combine information from multiple corpora. This approach was used previously in the state of the art narrative closure algorithm, to learn lexical information on events [5]. For this task, the VerbNet corpus will provide additional event specific information, which can help to reduce the sparsity within StoryCloze (Fig. 2) [7].

Overfitting was a recurring issue throughout our experiments, particularly in simpler models with fewer features (Table 1). We used L1 regularization to address overfitting of the training examples. We also experimented with L2 regularization, but found that error was higher with L2, possibly due to sparsity of the data.

## Future Directions

The limited size of the StoryCloze corpus provides minimal lexical information about the labelled events. As this information could potentially indicate event ordering relative to the narrative time frame, we seek to additionally train on the VerbNet corpus to integrate lexical information specific to verbs into the feature set.

Perhaps one of the most direct fixes to the StoryCloze corpus would be to add additional annotations. While the existing annotations in the corpus were added manually, our study reinforces the need for automatic event extraction and labeling to increase data set size, which would help to increase feature density and possibly balance classes as well.

Finally, our study highlights the difficulty associated with engineering feasible features to capture syntactic structure. With additional features (from VerbNet) and a substantial increase in the size of the annotated StoryCloze corpus, we posit that a deep-learning approach may be able to extract and select general features for sentence structure.

## Conclusion

Using a multiclass logistic regression classifier with hand-built features, we achieved 62% test accuracy on annotated 5-sentence short narratives, in which we found that features capturing broader sentence structure are more likely to be effective in predicting temporal relations across distinct sentences. We will extend our study to include event-specific information available in the VerbNet corpus, but posit that a deep-learning approach may be more successful in the identification of features for sentence structure.

## References

[1] W. Sun, A. Rumshisky, and O. Uzuner, “Evaluating temporal relations in clinical text: 2012 i2b2

Challenge.," *J. Am. Med. Inform. Assoc.*, vol. 20, no. 5, pp. 806–13, 2013.

[2] I. Mani, M. Verhagen, and B. Wellner, "Machine learning of temporal relations," *Proc. 21st …*, no. July, pp. 753–760, 2006.

[3] S. Bethard and J. H. Martin, "CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features," *Proc. 4th Int. Work. Semant. Eval.*, no. June, pp. 129–132, 2007.

[4] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," *Proc. Assoc. Comput. Linguist.*, vol. 31, no. 14, pp. 789–797, 2008.

[5] N. N. Chambers, "NavyTime: Event and Time Ordering from Raw Text," *Second Jt. Conf. Lex. Comput. Semant. (*SEM), Vol. 2 Proc. Seventh Int. Work. Semant. Eval. (SemEval 2013)*, vol. 2, no. SemEval, pp. 73–77, 2013.

[6] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories," *Proc. 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol.*, pp. 839–849, 2016.

[7] Schuler, Karin Kipper. "VerbNet: A broad-coverage, comprehensive verb lexicon." (2005).

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.