Detecting Temporal Relations of Events in Short Narratives

Delenn Chin, Kevin Chen

{delenn, kchen8} @stanford.edu

Introduction

The translation of ideas expressed in natural language to a computationally usable form remains a fundamental goal in NLP. Using an annotated corpus of short 5-sentence narratives, we developed a classifier for determining whether one event happens before, during, or after another event. With limited data, our classifier is able to achieve 62% accuracy in relation prediction.

Data

We use the annotated StoryCloze corpus, published by the Mostafazadeh group at Rochester, which consists of 300 5-sentence short stories, for a total of ~3,700 labelled event pairs. We focus our study to the classification using the provided temporal labels, {'BEFORE', 'OVERLAPS', 'DURING'}.

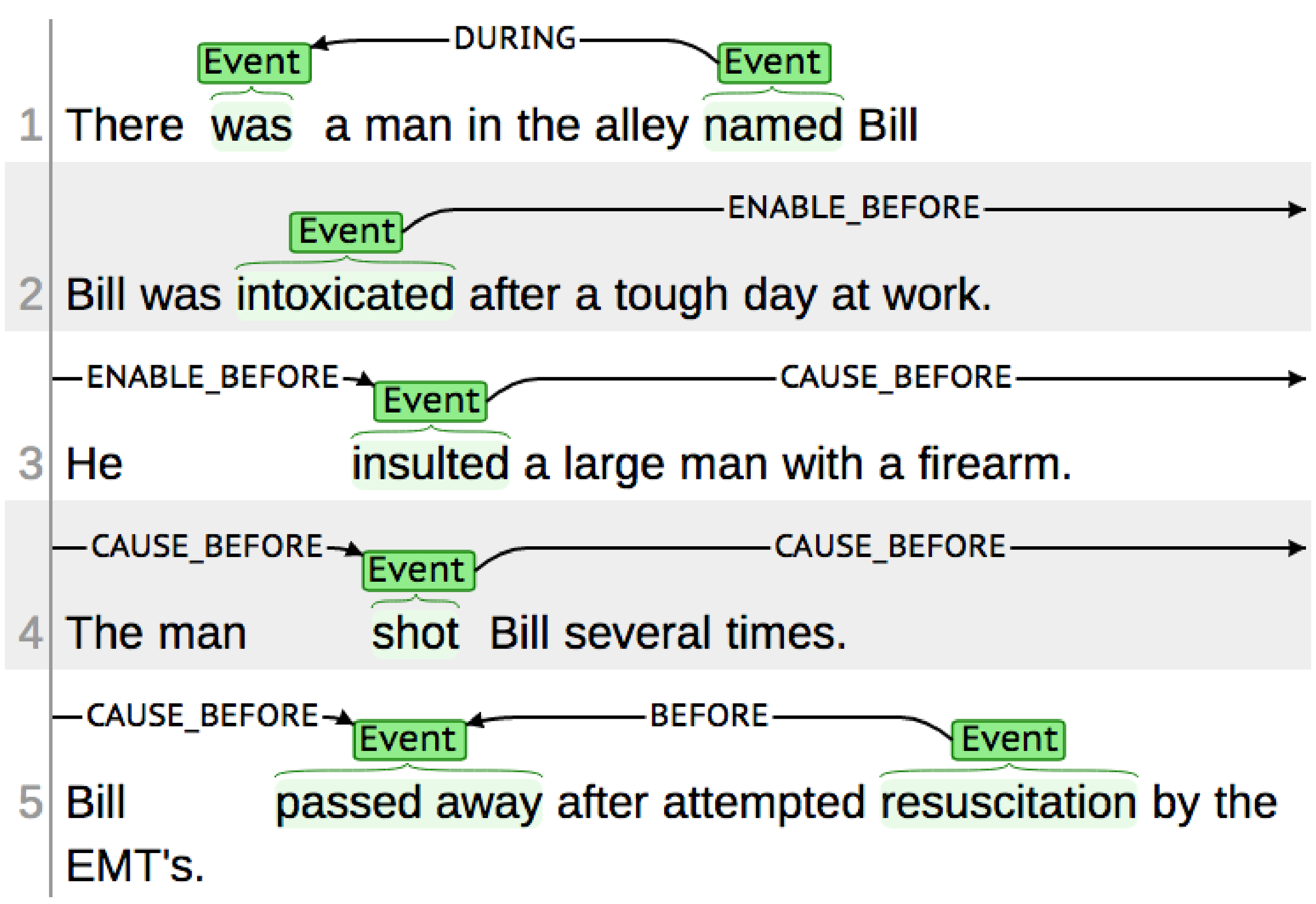

Figure 1: Sample story with annotations taken from StoryCloze corpus (Mostafazadeh et al. 2016).

Feature Selection

We experimented with features common to NLP tasks, as well as specific to temporal intuition.

- Events (word, lemma, synsets)
  - Tense
  - Ordering in document
- Number of tokens between events
- Part of Speech {uni, bi, tri}– grams around events
- L1 regularization

Model Selection

We framed the problem as a 3-way classification problem, where each pair of events is assigned a label from {'BEFORE', 'OVERLAPS', 'DURING'}.

Naïve Bayes

We first tried to use Naïve Bayes for multi-class classification, with the objective likelihood function

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

Event pairs were given the maximum likelihood class, and evaluated via accuracy. With unigram and bigram approaches, we achieved only accuracies of 50% and 53%.

Logistic Regression

We soon realized that temporal relations are often dependent on general sentence structure as opposed to the presence of tokens (with certain exceptions, ex. "after", "before", etc.), prompting a switch to multi-class logistic regression. Using the *sklearn* Python library, we maximize the Softmax regression function over all examples in the training set:

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

Results

- Models tend to have high training accuracy and low testing accuracy

Table 1: Model Accuracies and Sample Sizes

	Training F1-metric	Testing F1-metric
Naïve Bayes, w/ unigrams	0.94	0.50
Log-Reg, baseline	0.99	0.48
Log-Reg, w/ features	0.85	0.62
Num Samples	**2058**	**354**

Discussion

- Limited dataset and bias towards "BEFORE" relation makes classification challenging
  - Inherent bias in story telling, text sources toward temporal linearity
  - Overfitting to features specific to train set
  - Token specific features most heavily weighted in other classes
- Token count between events improved accuracy most - ~ 8%
- Token specific features sparse, as temporal relation less related to the actual words used

Future Directions

- Use VerbNet corpus to incorporate semantic features of events
- Increase data set size / number labelled stories, try to reduce bias

References

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.

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