A Novel Approach to Event Duration Prediction

Divye Khilnani, Pranav Khaitan and Ye Jin

1. Abstract

Durations of Events, play a pivotal role in temporal reasoning problems. Accurate estimates of time periods of events can help us obtain better solutions for several time related tasks such as sequencing events and identifying temporal relations between them.

In this paper we explore the application of supervised and unsupervised machine learning techniques to predict coarse-grained as well as fine-grained estimations of event durations. We use linguistic, contextual, temporal and generic features to train our classifiers. The paper presents both, an analysis of the features used and the results obtained on the Time Bank Corpus. Moreover, our result for Naïve Bayes Classification is better than that presented by Feng pan et al [1] by 4.86%.

2. Introduction

Predicting duration of events is an intriguing problem that can potentially be solved using various natural language processing techniques. For instance, consider the following statement that occurs in an article:

Liverpool will be playing versus Inter-Milan this Friday.

From the above sentence we can deduce that the event it refers to is the Football match between the two teams. Identifying the duration of this match can help us to make other meaningful conclusions. Consider another statement that is also present in the same article:

The tournament ends on Monday.

Moreover, let us assume that we have identified the duration of the match to be in hours while the duration of the tournament to be in weeks. Thus by placing a bound on the time period of these events we can conclusively identify the temporal relation between them, which is that the match is a part of the tournament.

However, associating a duration with an event is non-trivial because the same event can have different bounds in different contexts. For instance consider the following two examples:

James watched a movie.

James watched the birds fly.

In both of the above sentences, the event is identical, which is 'to watch'. However, watching a movie is an activity that would generally last for couple of hours while watching birds fly would ideally end in a few minutes. Thus in order to correctly associate a duration label with the event 'to watch', we have to consider other non-temporal features such as the object of the sentence. Such linguistic features can provide insightful information about the event, thus enabling us to improve our prediction.

Moreover, specifying the duration of an event is often subjective. In a related study undertaken by Feng Pan et all[2], the researchers asked participants to specify the duration of events identified from the Time Bank corpus. It is interesting to note that the Human Agreement Precision (percentage of events which received the same duration label from all human annotators) was 87.7% in coarse-grained classification and as low as 44.4% in fine grained classification.

We initially parsed the TimeBank corpus which comprises of news articles to identify events. The corpus was initially split into two disjoint sets; the train dataset and the test dataset. For each event identified, we create a Temporal Event object as well as a DurationPredicateNew object, which are used to extract all relevant features corresponding to that event. We then map both these objects to a DataItem object which consists of the final features list. The DataItemGenerator Class generates all the data items. Once the list of

training and test DataItems has been created we train and test the corresponding classifiers. We initially used K-Fold Cross Validation to test our features. We have also implemented two feature selection techniques which are Mutual Information and Chi-Square. Apart from using standard performance metrics such as Precision, F1-Measure and Recall we also implemented the Kappa Statistic as well as Inter Annotator Agreement Metric, as explained by Feng Pan et al[2].

In section 3 we present the previous work that has been done in this area. We then give an overview of the supervised and unsupervised machine learning techniques in section 4 and 5 respectively. We present our analysis on various features in section 6. Section 7 explains the feature selection techniques used by us. Section 8 further explains the evaluation metrics used by us. Finally we present our Results, Future Work and Conclusion in sections 9,10, and 11 respectively.

3. Previous Work

Recognizing events and their durations has attracted a lot of interest in recent areas. This interest mainly comes from the requirement of temporal information in question answering systems. Pustejovsky, et al [3] introduced TimeML as a specification language for events and temporal expressions in natural language. TimeML uses XML to annotate the core elements for temporal analysis.

There has been a good amount of work done on analyzing news stories [4,5]. It has been observed that the events are often presented in a different sequence than the sequence in which it occurred. This makes it quite difficult to order events. It becomes even more complex when we are dealing with multiple events. For example, a news article may talk about the fact that the inauguration of President Obama was taking place at the same time when the stock markets were performing very badly. It is important to recognize that the second event is a longer event here and the time period when the first event took place was a subset of that of the second event. Filatova et al [6] did some work on breaking news events into their constituent events and assigning timestamps to them. They achieved a performance of 52% compared to humans and concluded that a list of machine learning techniques is required for time-detection.

Recently, Pan et al [1] have used machine learning techniques to estimate event durations. They use syntactic features to do course grained analysis and classify events as either longer than a day or shorter than a day. They could not come up with any features other than syntactic features and the event itself which improves performance.

Lapata, et al [7] used a probabilistic model to infer temporal relations within sentences. Their model gives an inference accuracy of 70.7%. An interesting thing about their experiment is that they draw semantic information without using any semantic annotations.

Chambers, et al [8] use many imperfect event attributes effectively to get temporal information about the event. They use these features to determine ordering between events and many of these features can also be used for the task of mapping event to their durations.

4. Supervised learning

Given that we had sufficient number of annotated data, our first approach was to use supervised learning techniques for classification. We carried out two kinds of classification tasks.

Course grained Classifier:

In course grained classifiers we classified events into two buckets based on a pivot duration. For example, if we consider hour as a pivot duration, then all events which take place for a period great or equal to hour will be classified as “Greater than hour” while events which take place for a period less than an hour will be classified as “Less than hour”. We created a different binary classifier for each event period such as minute, hour, day, week and month.

If we consider only the average accuracy, the classifiers minute and month had a relatively higher accuracy because they had a skewed distribution. However, the day classifier would likely have a greater utility because it is often more important to find out if an event took place for more than a day or less than a day than

it is important to know if it occurred in 7 days or 12 days. For example, if there is a meeting between the Presidents of the country it is often difficult to tell if the meeting will last less than a day or greater than a day. However, it is very easy to tell that the meeting will last greater than a minute and less than a month. Similarly, it is difficult for a day classifier to say if a local tennis tournament will be played for less than a day or greater than a day while other classifiers would do relatively well at this task.

Fine grained Classifier:

Even though course grained classifiers are quite useful, they generalize the durations to a great extent which leads to loss of significant information. For example, an hour classifier would classify a soccer match and a world war both to be greater than an hour and we do not get much value out of this classification.

We therefore switch to the task of fine-grained classification where we classify events into one of the eight classes - seconds, minutes, hours, days, weeks, months, years, decades. These classes are particularly useful when trying to order events and determine if an event duration is smaller or larger than the other one.

We use a number of machine learning techniques and analyze their results to come up with interesting observations as to how the different techniques behave with each of ours classifiers.

4.1. Naïve Bayes

The naïve Bayes model is based on the assumption of conditional independence between all features. Despite its simplicity, it has been shown to be a very powerful model. A major advantage of using this technique is that it is relatively fast to compute. Unlike svm or logistic regression which take multiple features into account at the same time, this classifier treats each feature independently. Naïve Bayes also tends to do less overfitting compared to logistic regression. Therefore, even though its training accuracy is often less than logistic, its test accuracy is higher than all other models we tried.

Using the definition of conditional probability and conditional independence assumption, we can get the following formula:

$$P(C \mid F_1, F_2, \ldots, F_n) = \frac{P(C)\, P(F_1 \mid C) P(F_2 \mid C) \ldots P(F_n \mid C)}{P(F_1, F_2, \ldots, F_n)}$$

Since the denominator is constant for all classes, we just have to find the class C which maximizes the expression $P(C) \prod_i P(F_i \mid C)$.

During implementation, we use log probabilities because addition is much faster than multiplication and it also avoids floating point precision problems.

We calculate the probabilities P(C) and $P(F_i \mid C)$ during training as follows:

$$P(C) = \frac{n(C)}{N}$$

$$P(F_i \mid C) = \frac{n(F_i, C) + 1}{n(C) + n_f}$$

Where N is the total number of training examples, $n(C)$ is the total number of training examples of class C, $n(F_i, C)$ is the total number of training examples of class C which also contains the i^{th} feature and n_f is the total number of features in our training set. We used add-one smoothing to give non-zero probabilities to instances which were not present in our training set.

4.2. Logistic Regression

We also analyze the performance of our feature sets with logistic regression which is a discrimative model. We use the following formula for logistic regression where x is our feature set and θ is the coefficient vector which we learn from the training set. An example is classified as 0 or 1 if the value of $h_\theta(x)$ is less than or greater than 0.5 respectively.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Logistic regression does not hold any strong assumptions unlike Naïve Bayes or GDA.

However, the drawback of this algorithm is that it is not very stable and it requires a lot of training data to give good results. Since our training set size is modest, this model is often outperformed by Naïve Bayes.

4.3 Maximum Entropy classifier:

The maximum entropy classifier is a feature based classifier where we use the training set to assigns weights for each of the features and determine the conditional distribution. During the testing we calculate the probability of the data item belong to each class as follows and then choose the class which has the highest probability. Since the denominator is constant for all classes, we simply calculate the numerator and choose the class which maximizes this value.

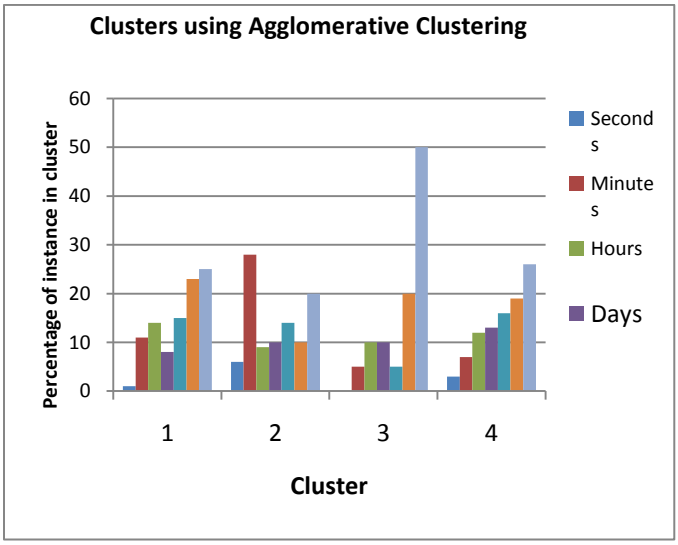
$$P(c \mid d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c, d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c', d)}$$

5. Unsupervised learning

As is often the case in the machine learning world, it is sometimes difficult and expensive to get a lot of labeled data. Even for temporal events, there isn't a lot of annotated data to train our model on. However, there is tons of un-annotated data which we could use to our advantage. We experimented with clustering algorithms to see if our features can actually be used to group examples intro different clusters even when we don't have the labels during training.

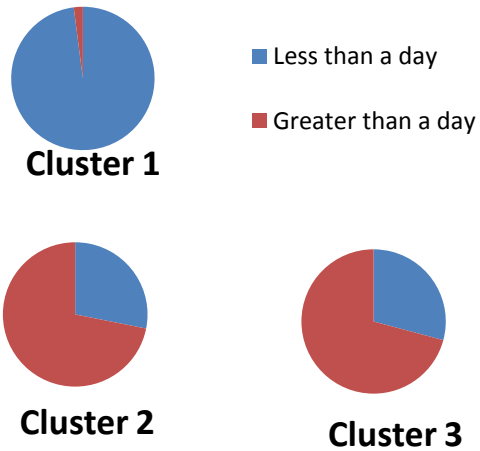
5.1 Agglomerative Clustering

The first method which we tried was Agglomerative Clustering which is a very naïve clustering technique. We used the cosine distance as a measure the distance. Among single and complete link, complete link performed better. After combining our observations with our theory, we found that complete link does better because it takes into account the association of a node with all the nodes in the cluster rather than its proximity to only the closest node in the cluster. We can see the performance of our clustering algorithm using agglomerative clustering below. Clusters three and four contain mostly months and years and they seem to be quite useful in classifying. However, we cannot find any cluster with a higher percentage of short duration events.

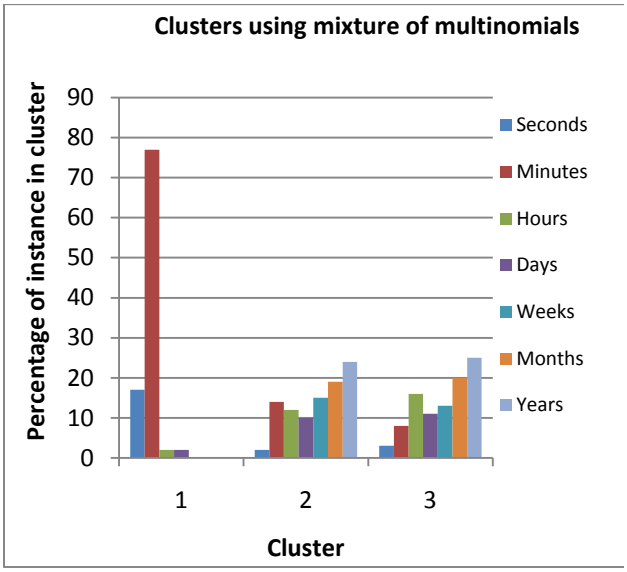


5.2 Mixture of multinomial clustering

Due to limitations of agglomerative clustering, we decided to take up mixture of multinomial clustering which is a probabilistic model. This is a generative model, where we assume each cluster to be like a separate class. This model works quite well for our data. Not only is it significantly better than agglomerative clustering, it gives us very good clusters. The pie-charts below give an idea about the homogeneity of our clusters. The first cluster almost entirely contains events which occur for less than a day. The second and third clusters and not very pure but more than two-thirds of the events in these clusters have duration greater than a day.



We also get a detailed idea about our clusters by doing fine grained analysis on them. The graph below shows the number of elements for each bucket in each of the three clusters. We can observe that it is often difficult for our clustering algorithm to differentiate between hourly and daily events. Each of the three clusters contain approximately the same proportion of hourly and daily events. However, our clustering algorithm does a good task of differentiating months and years from events with smaller durations. The first cluster doesn't contain any event belonging to months or durations while the second and clusters have more of these elements.



6. Feature Analysis

In order to train our classifiers we used several lexical, syntactic, semantic and generic features. For an objective comparison of our features we have defined the baseline measure as the results obtained when we only incorporate the event name itself as a feature. Thus if the given sentence is:

The President is attending the conference.

The event will be 'attending' and the baseline feature will be the event name which is 'to attend'.

Presented below is our analysis of the gains obtained by incorporating different features.

6.1 Linguistic Features:

As explained in the earlier section, our events have been extracted from sentences of news articles. Thus, the rationale behind including linguistic features was to capture the semantics of the sentence and the context in which the 'event word' was being used. The linguistic and lexical features we considered are:

Subject – Object: The duration of an event is often governed by the participants of that event, which in grammatical terms can be described as the subject and the object, if the event is expressed as verb. For instance consider the event 'to see'. We have two sentences which are:

Nina saw the advertisement

Nina saw the play.

While the subject in both the sentences is Nina, the object in the first sentence is advertisement while the object in the second sentence is play. Also, from common knowledge we know that a play is much longer than an advertisement. By training our classifier to use this information while predicting event durations, we can get a significant boost in performance.

We obtain the subject and object feature values by generating a parse tree and extracting the head word of the subject and object of the sentence. The table below gives an overview of the accuracies obtained in the coarse grained classification task.

Classifier	Baseline	Baseline + Subject-Object
NB	71.97%	71.12%
LR	60.08%	72.61%
MaxEnt	71.97%	72.82%

Base Verb Lemmatization: A given verb can be used in multiple ways in a sentence based on the context. For instant the verb 'to eat' can be used as eat, eating, is eating, will be eating, ate, has eaten and so on.

He ate lunch.

He will be eating lunch.

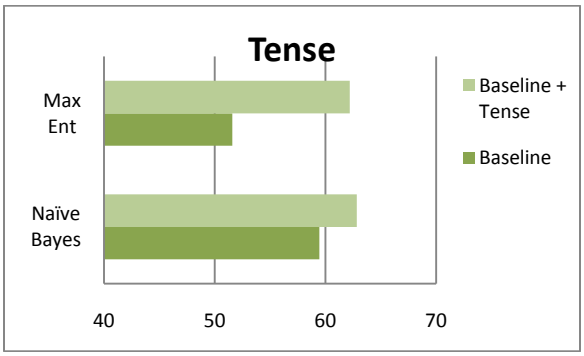
By mapping these multiple verb forms to a base verb and incorporating that as a feature we enable the classifiers to find common patterns amongst related events. Moreover, by increasing the number of shared features amongst similar events, we are reducing the sparseness of the training and test data.

Tense: Another syntactic feature which we considered was the tense of the event. Tense can play an important role in distinguishing between ongoing and future events. If the event has already begun it is likely to have a shorter duration than an event which will be starting in the future. Moreover, the tense can also help us to better understand the semantics of the sentence. Consider the following two sentences:

Nick will play football in the evening

Nick has been playing football for the past ten years.

In the first sentence, the verb 'to play' is in the future progressive tense while in the second sentence it is in the present perfect progressive tense. This differentiation between the tenses helps us too understand that the first sentence is referring to a game of football and hence its duration will only be a couple of hours, while the second sentence is referring to Nick's career in football and in this context the event spans over years. Incorporating the tense feature significantly boosted the approximate agreement measures of the fine grained classifiers.



Part of Speech: Part of Speech Tagging (PoST) can help us to better capture the semantic meaning of a word based on its definition and its contextual usage in the sentence. We perform PoST not only for the event word, but also for other words in the sentence which we include as features for a given event. PoST helps us to better understand how these words relate to the event. Moreover, sometimes the event might not be a verb itself. For example consider the following sentence:

The move has been anticipated by the Ireland Police.

In the above sentence, the event, which in this example, is 'move' is not a verb but a noun. Move when used in the context often refers to political or strategic actions whose durations span over a period of months or years. By incorporating PoST we can differentiate between another usage of move such as 'moving the chair', which is an event that will last for only a few minutes. Similarly PoST can help us to differentiate between different senses of a word. For instance the word 'dogged' can be used as an adjective or as a past tense word [10]. Thus, by incorporating PoST our classifiers are better equipped to handle words that demonstrate polysemy.

Sentential Dependencies: Apart from the subject and object an event may have several other dependencies such as adverbial modifiers, adjectives, relative clause modifiers and quantitative phrase modifiers[13]. By incorporating these dependencies we can capture relevant temporal information, that can help us to improve our prediction accuracies on the coarse grained as well as fine grained tasks. For instance, one of the sentences in the corpus contained the phrase 'slowly enticing the youth'. Here, slowly is an adverb which gives further temporal information about the event which is 'to entice'. We can expect events with 'slowly' adjective to have longer durations than events without it. Moreover the duration will be much smaller than events which have qualifiers such as quickly or rapidly.

Most dependencies are more complex than the one presented above. Consider the following sentence:

The art of socializing is also experiencing a change where Net/virtual relationships are fast overtaking or becoming parallel with the normal human relationships.

In the above sentence one of the events is change which itself is a noun and has the relational clause modifier 'overtaking or becoming parallel with'. Thus we can conclude that the change involves 'overtaking' something, which is a time consuming process.

We incorporated the dependency feature in two stages. Initially we only incorporated the words which are dependent on or dependents of the 'event word'. This gave us a substantial boost in the prediction accuracy. We then also incorporated the actual relation that the word shares with the event word. Doing so further improved our results. We believe this occurs because by incorporating the relations as well we are able to better capture between the context of their usage and hence the semantics of the sentence. The table below summarizes the improvement in prediction accuracies obtained in coarse grained classification:

Classifier	**Baseline**	**Baseline + Subject-Object**
NB	71.97%	73.03%
LR	60.08%	74.31%
MaxEnt	71.97%	73.46%

6.2 Named Entity Recognition of Subject and Object:

The rationale behind adding this feature was to differentiate between events that relate to individuals versus those that relate to groups. While performing Named Entity Recognition we classify entities into four groups which are: PERSON, ORGANIZATION, LOCATION and OTHER. The fact that the subject or an object is an organization can significantly affect the duration of the event. For instance, one of the sentences in the data set contained the following phrase:

define the role of United Nations.

Here the event is to define while the object is United Nations, an organization. By specifying that the latter is an organization, we can capture the intuition that defining an organization's role is a complex task which is likely to take more time as compared to defining the role of an individual. By including the NER feature we got a boost of nearly 1% in the inter annotator judgment metric for the fine grained temporal classification task using the naïve bayes classifier. Presented below are the improvements we obtained in prediction accuracies for the coarse grained classification experiment using NER:

Classifier	**Baseline**	**Baseline + Subject-Object**
NB	71.97%	71.97%
LR	60.08%	62%
MaxEnt	71.97%	73.25%

It is interesting to note that for both the Naïve Bayes Model and the Maximum Entropy model, the gain obtained in prediction accuracy is greater by incorporating the NER values of the Subject and Object rather than incorporating their actual values. One possible reason, for this could be that since the Naïve Bayes and Maximum Entropy are probabilistic models the conditional probabilities assigned to various classes given particular feature values are low due to the exponentially large number of possible values the subject and object can take. NER on the other hand helps to better distinguish between group and individual events and hence is a more useful feature.

6.3 Web Counts:

Apart from using lexical and semantic based features we also decided to use features which accurately capture the trends in usage of the 'event words'. One such feature is web counts which is the total number of hits obtained for a particular query. The queries were generally of the form X*Y where X was a phrase containing the event word and Y was a temporal phrase such as hours, years, in

April, and by next week. Some of the sample queries are given below:

Washington user for * hours
spent * weeks returning
People predicted for * years
was flaring for * months

In order to obtain the webcount feature we first extracted all the queries relating to a particular event. We then created buckets for each of the following temporal classes, which are seconds, minutes, hours, days, weeks, months, years, and decades. The queries were then mapped to their corresponding buckets. Each query was associated with a weight which represented the number of hits obtained for that query using the yahoo search engine. Thus for each event we were able to obtain a distribution of its queries over the various temporal labels mentioned above. We then selected the bucket having the maximum number of hits and added that as a feature.

The webcount feature boosted the prediction accuracy of the coarse grained classifiers by nearly 1% and that of the fine grained classifiers by 2%.

6.4 Generic Features:

We also incorporated a number of generic features. While some of these features were useful, the gains obtained were not as significant as those obtained from other features mentioned above. For the sake of brevity, we will only present a short overview of these features:

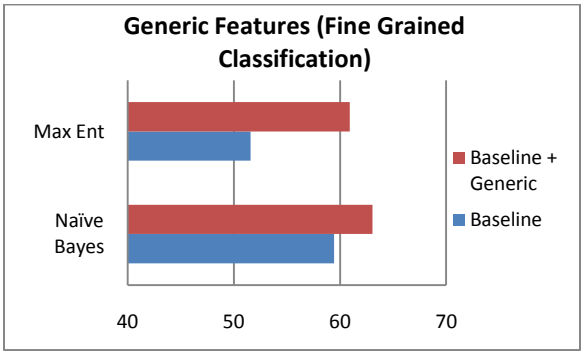
Modality: Modality is used to express possibility or necessity. Modal clauses often contain words such as could, would, and must. The objective of including modality as a feature was to capture the intuition that events that are likely to occur with some probability will have longer durations than an event which will definitely occur. For instance consider the sentence:

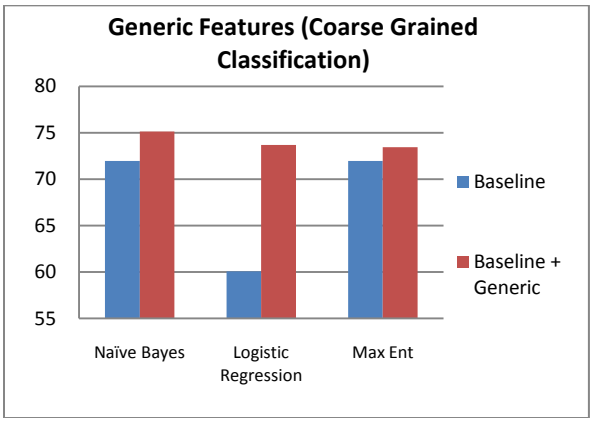
The easing of tensions could bring several soldiers home.

In the above sentence, the use of the modal word 'could' indicates that the event of bringing the soldiers home can happen a few days or even in a few months.

Aspect: The grammatical aspect of a verb defines the temporal flow (or lack thereof) in the described event or state. For example, the present tense sentences "I swim" and "I am swimming" differ in aspect (the first sentence is in what is called the habitual aspect, and the second is in what is called the progressive, or continuous, aspect). In our study we distinguish between progressive aspect such as challenging and perfective aspect such as challenged. The intuition is that progressive events are likely to last longer than perfective events.

Class: The class feature captures the nature of the event itself. The various classes that we divide events into include reporting, action, state, and occurrence. By grouping events which are fundamentally similar, such as those denoting actions we hope to be able to make better predictions about the event durations as events falling in the same category will have similar time periods.





The graphs above chart the gains obtained in the prediction accuracies of the coarse grained classifiers and approximate agreement measure of the fine grained classifiers.

6.5 Hypernyms:

Here we try to capture the inner class of the word and use it as a feature to subgroup the events. For example, talk, chat, speak can be grouped together and have the same hypernym as communication. Naturally, events have the same hypernym will have similar durations in the corpus. Here, we used WordNet to query the synset of a particular word and search all the way down the hypernyms chain. However, to get the hypernym, we encountered the problem of word understanding, the problem of word disambiguation. For example,

COMMUTE

1. *travel back and forth regularly, as between one's place of work and home*
2. *exchange or replace with another, usually of the same kind or category*

This can lead to different hypernyms in different time durations. Instead of a rather sophisticated analysis of word disambiguation, we decided to go for an easier approach. For each word, we will go with the first and most common meaning. Then we capture the different meanings by applying hypernym to the subject and object of the event as well. The result shows that the accuracy goes up for extracting the hypernym value of an event. Presented below are the improvements we obtained in prediction accuracies for the coarse grained classification experiment using Hypernym:

Classifier	Baseline	B+H	B+H(S/O)
NB	71.97%	71.09%	70.4%
LR	60.08%	71.20%	73.2%
MaxEnt	71.97%	72.39%	73.46%

Note: B+H indicates Baseline + Hypernym, B+H(S/O) means that Baseline + Hypernym with Subject Hypernym and Object Hypernym.

6.6 Contextual Features:

For a given event, the local context features include a window of n tokens to its left and n tokens to its right, as well as the event itself. A token can be a

word or a punctuation mark. For each token in the local context, including the event itself, three features are included: the original form of the token, its lemma (or root form), and its part-of-speech (POS) tag. Presented below are the improvements we obtained in prediction accuracies for the coarse grained classification experiment using Context:

Classifier	Baseline	B+C1	B+C2
NB	71.97%	74.78%	71.66%
LR	60.08%	73.26%	72.19%
MaxEnt	71.97%	73.89%	72.40%

6.7 Report Feature:

We have noticed that there are many reporting verbs in the corpus. For example: complain, confirm, deny, doubt, estimate, explain, propose, remark. Further analysis shows that these reporting events have similar time durations. The large amount of reporting events makes us decide to add one more feature, the report feature. It is a simple feature. We white list a group of Reporting events, and whenever the revent is in the list, a report feature will be added. This report feature helped us to increase our performance further.

7. Feature Selection

The number of features we have collected for a typical binary classifier is up to 5,600 based on the 2,132 events in the training sentences. To train with all the features indicate a performance issue as well as some scalability problem. What's more, the result shows a certain number of features are null in a considerable large amount of cases, which might lead to higher variance. Therefore, not all variables that are selected are likely to be necessary for accurate discrimination and including them in the classification model may in fact lead to a worse model than if they were removed. We implemented two algorithms to achieve feature selection.

7.1 Mutual Information

The mutual information of two discrete random variables X and Y is defined as follows:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right)$$

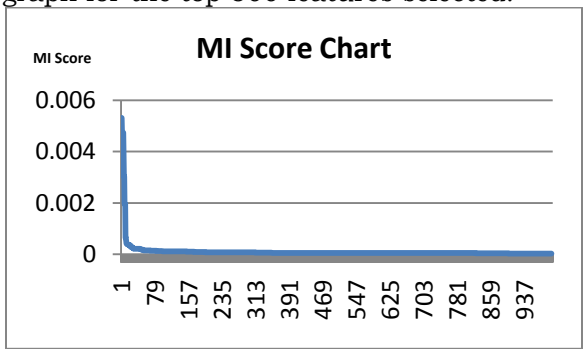
This formula measures the information that X and Y share: it measures how much knowing one of these variables reduces our uncertainty about the other.

In our case, for each class C and each feature F, the function can be defined as follows:

$$I(C;F) = \sum_{f \in \{1,0\}} \sum_{c \in C} p(c,f) \log\left(\frac{p(c,f)}{p_1(c)p_2(f)}\right)$$

The above equation shows that the features are binary, either existing or not. And for each feature, we calculate its mutual information score with the associated classes. We sort the features according to their score and only the top K numbers will be selected to train and test.

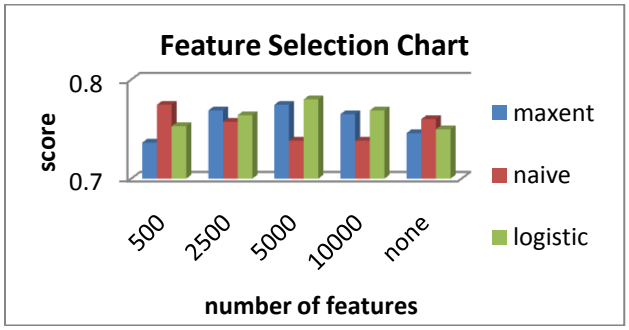
Presented below is the MI-Score graph for the top 500 features selected:



The above chart shows that MI score drops quickly after the first 100 features.

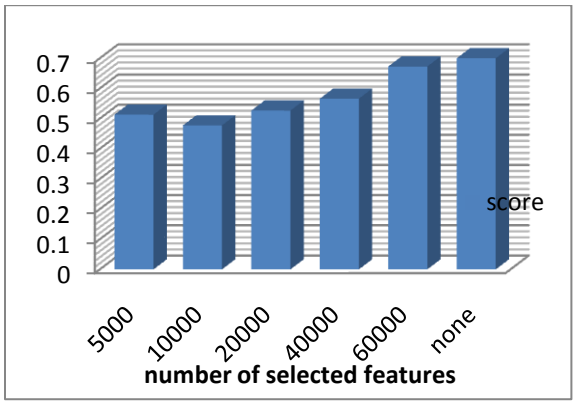
Presented below is a chart showing the accuracy score for the top K features selected.

The curve for Logistic Regression Classifier and MaxEnt classifier score is lower at the two sides and higher in the middle. This shows that the performance of the feature selection find a local optima for the score, which is around 5,000 features out of more than 10,000. But as we see that since the MI score drops rapidly the increasing feature doesn't improve the score much. which is around 5,000 features out of more than 10,000. But as we see that since the MI score drops rapidly, increasing feature doesn't improve the score much.



Thus, from 500 to 10,000 features, the score varies only within the range of 5%.

For fine grained training result, feature selection shows a monotonous drop in keeping fewer features. Below is the chart of fine grained classification with feature selection:



One possible reason for the drop in score is that the number of features for the fine-grained are not sufficient and more features should be extracted and explored further.

7.2 X^2 Feature Selection

This selection algorithm is similar to mutual information selection in terms of implementation. Statistically speaking, x^2 is a measure of the independences of two variables. If the score is higher, it means that they are more independent. In our cases, we use this score under the assumption that the feature and the class are independent. The equation of x^2 score:

$$x^2(F, C) = \frac{N(N_{11}N_{10} - N_{10}N_{01})^2}{(N_{11} + N_{01})(N_{11} + N_{10})(N_{10} + N_{00})(N_{01} + N_{00})}$$

Note: N is the total number of training sentences. N_11 is the number of co-occurrence of the feature F and the class C. N_10is the number of sentences contains the feature F but is not in class C. N_01is the number of sentences in clas C but F. doesn't contain feature F. The result for x^2 is similar to MI but performing slightly worse. Therefore, we are not discussing them again. One thing to note is that the highest x^2 score is 27.17 and it also has a steep drop. One reason for the slightly worse overall performance is that for duration predicates, the class and features do not fulfill the assumption that it is independent. N_00 is the number of sentences not in C and doesn't contain feature.

8. Evaluation

To evaluate the performance of our classifiers, we implemented four methods, the traditional score: precision, recall, F1 and kappa statistic measurement.

8.1 Precision, Recall, F1

Precision indicates the false positives and recall shows the false negatives.

For a result set, we have

tp(true positive)	fp(false positive)
Fn(false negative)	tn(true negative)

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

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

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

8.2 Inter-Annotator Agreement:

Although the above scores is a straightforward indicator of how well our classifiers are performing, the classes among themselves are rather complicated. For example, if an hour event is classified wrong, it can be classified as day event or minute event. In reality, as we know that if an event happens within an hour, it is likely to be more related to minute events rather than hours events. However, the above measure will give them the same score. Therefore, we need a more detailed quantitative measurement of the agreement for each event. Here, we used the kappa statistic(Carletta, 1996) for the measurement:

$$k = \frac{P(A) - P(E)}{1 - P(E)}$$

P(A) Class Agreement:

In our cases, P(A) is the agreement between two classes. It is computed as follows :
Our data model is represented by normal distributions. First, all duration classes are represented in seconds. Further, because duration ranges from seconds to decades, we applied natural logarithmic scale to is duration. For a particular class, we will build a normal distribution; the mean of the distribution is the average time of the class and the bounds are each 1.28 standard deviations from the mean. With this data model, the agreement between two normal classes can be defined as the overlapping area between two normal distributions.

P(E) Expected Agreement:

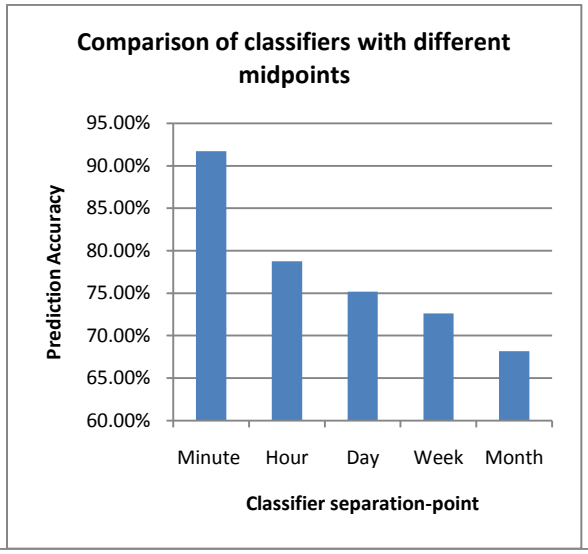
The probability that the classes agree by chance is that there is a global distribution from the training data. The distribution is drawn from the duration ranges for all the events. This is a global distribution which is calculated as follows:
First is to build a histogram. The x-axis is the mean values in the natural logarithmic scale. The y-axis represents the number of annotated durations with that mean.
Then is to compute the distribution of the width of the durations. The x-axis is duration width in the natural logarithmic scale. The y-axis represents the number of annotated durations with that width.
Here we used the score of Pan et al., 2006's result. We set P(E) = 0.15.

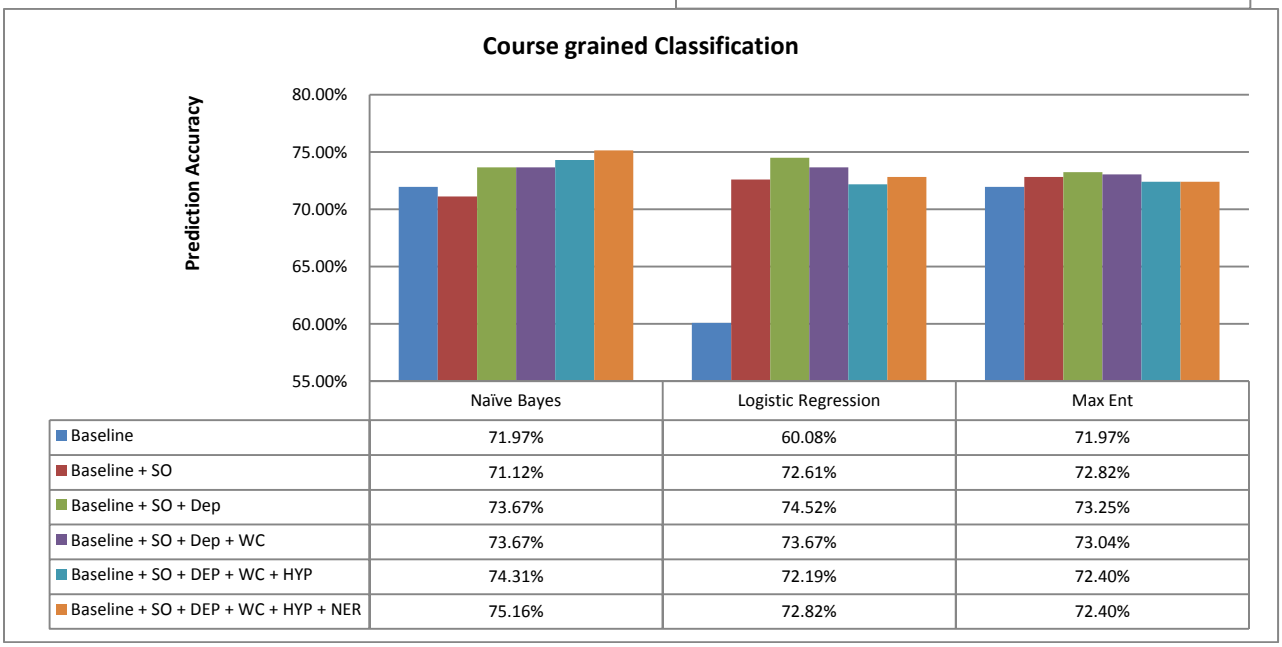
Approximate Agreement:
Human agreement on assigning a finegrained temporal unit to a class is shown to be only about 44.4% [1]. This indicates that it is very difficult to evaluate the assigned temporal units based on this measure. We therefore evaluated our classifiers on an alternate measure called "approximate agreement". According to this measure, we consider a prediction to be correct if it is the same as the gold standard assignment or if it is adjacent to the original assignment. For example, if we had a gold standard duration value of days, then we would consider hours, days and weeks to be correct. A similar measure has been used by Pan, et al [1] for evaluating their classifiers.

9. Results:

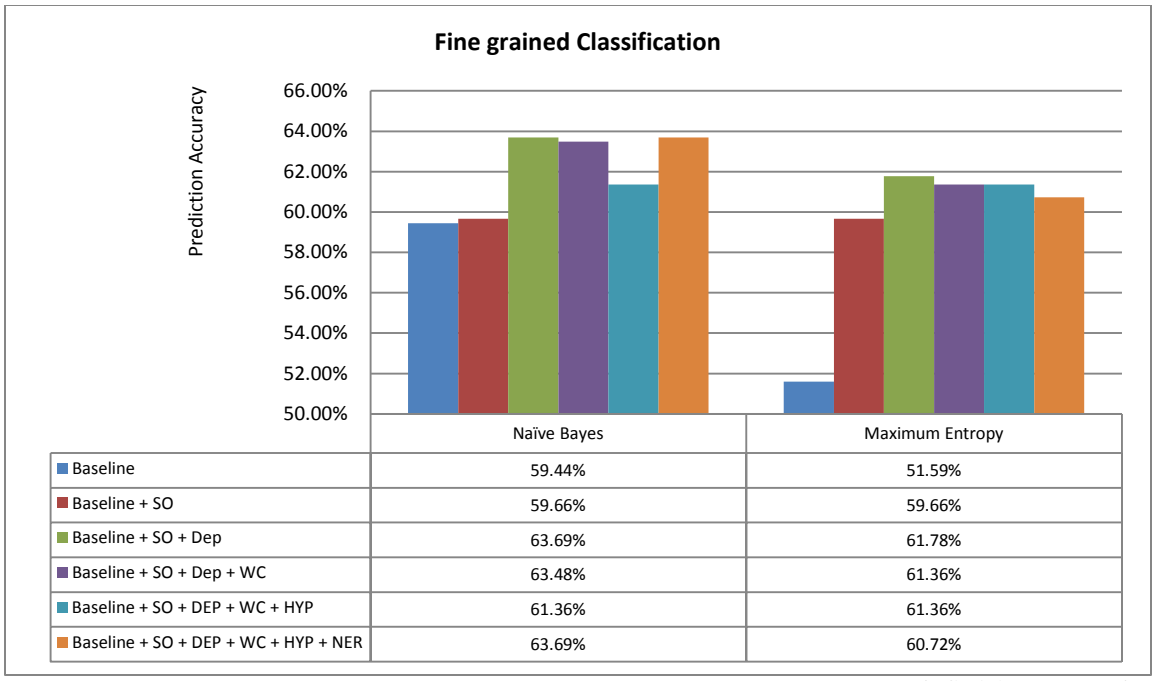
Presented below are the results obtained by combining the various features explained above. For convenience we will be using the following acronyms to refer to features.

Feature Name	Acronym
Subject-Object	SO
Dependencies	Dep
Web Counts	WC
Hypernyms	Hyp
Named Entity Recognition	NER





	Naïve Bayes	Logistic Regression	Max Ent
Baseline	71.97%	60.08%	71.97%
Baseline + SO	71.12%	72.61%	72.82%
Baseline + SO + Dep	73.67%	74.52%	73.25%
Baseline + SO + Dep + WC	73.67%	73.67%	73.04%
Baseline + SO + DEP + WC + HYP	74.31%	72.19%	72.40%
Baseline + SO + DEP + WC + HYP + NER	75.16%	72.82%	72.40%



	Naïve Bayes	Maximum Entropy
Baseline	59.44%	51.59%
Baseline + SO	59.66%	59.66%
Baseline + SO + Dep	63.69%	61.78%
Baseline + SO + Dep + WC	63.48%	61.36%
Baseline + SO + DEP + WC + HYP	61.36%	61.36%
Baseline + SO + DEP + WC + HYP + NER	63.69%	60.72%

We trained the coarse grained classifier using different separation points, which were minutes, hours, days, weeks and months. For each separation point the training data was labeled as less than that time label or greater than that time label. Hence if the separation point is days then the data items were labeled as less than a day and greater than a day. As anticipated the prediction accuracy decreases as the separation point increases from minutes to months. This occurs because if the separation point is low, such as minutes, most of the training examples will receive the label greater than a minute and hence the baseline accuracy itself will be high. Moreover, the temporal clues for differentiating between smaller separation points such as less than an hour and greater than an hour are more effective than differentiating between larger separation points such as greater than a month and less than a month. There are several events such as elections, war and court proceedings that could end in a month or could extend to several months. Thus the time periods of longer duration events are not as definitive as shorter duration events.

The highest coarse grained classification accuracy achieved by us is 75.16% while the highest fine grained classification accuracy achieved is 63.69%. As explained in the feature analysis section, certain features such as dependencies, and named entity recognition provide higher gains than other features such as web counts and hypernyms. The table below presents the Precision, Recall and F1 measures obtained by us for the coarse grained classification task when we consider all the features.

Classifier	Precision	Recall	F1
Naïve Bayes	74.01%	70.37%	0.73
Logistic Regression	74.68%	74.20%	0.75
Max Ent	73.39%	70.03%	0.72

Conclusion and Future Work:

In this paper we have presented an analysis of the application of various NLP based features for the purpose of predicting

event durations. The result obtained by us show that by incorporating the right features we can obtain a gain of nearly 15% in the coarse grain duration prediction task and a gain of around 10% in the fine-grained prediction task. The table below compares our results (prediction accuracies for the coarse grain classification task and approximate agreement for the fine grain classification task) with those obtained by Feng Pan et al [1] and the Human Agreement Score (an upper bound on the prediction accuracies).

Classification Task	Our Results	Feng Pan et al	Human Agree-ment
Coarse Grain	75.16%	70.3%	87.7%
Fine Grain	63.69%	65.8%	79.8%

Predicting time durations of an event is an important research topic as it can be further used for temporal reasoning problems such as event sequence ordering. Event duration prediction could also be used to improve the results provided for question answering systems, especially with respect to temporal based questions.

The next significant step in temporal duration mapping would be to analyze the neighboring sentences to be able to predict the durations with a higher precision. However, this will lead to an exponential rise in the number of feature combinations and it is relatively expensive to get a training set large enough to be representative of all these features. We therefore consider unsupervised learning to be the next best solution to the problem. We performed clustering on our data and observe that it gives rise to some very homogenous clusters which can be very useful in estimating event durations. Our next step would be to collect millions of events by crawling the web and performing clustering on this large corpus. After forming clusters, we can use our relatively small set of labeled sentences to assign labels to each clusters. Another interesting step in this research would be to develop features which learn event durations in a domain independent manner.

11. References:

[1] Feng Pan, Rutu Mulkar-Mehta, and Jerry R. Hobbs. 2007. "Modeling and Learning Vague Event Durations for Temporal Reasoning." In Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI), Nectar Track, pp. 1659-1662

[2] Feng Pan, Rutu Mulkar, and Jerry Hobbs. Learning Event Durations from Event Descriptions. Proceedings of Workshop on Annotation and Reasoning about Time and Events (ARTE'2006), ACL'2006.

[3] J. Pustejovsky, B. Ingria, R. Sauri, J. Castano, J. Littman, R. Gaizauskas, A. Setzer, G. Katz, and I. Mani. 2005. The Specification Language TimeML. In I. Mani, J. Pustejovsky, and R. Gaizauskas, (eds.), The Language of Time: A Reader. Oxford University Press.

[4] C. Schokkenbroek, 1999. News Stories: Structure, Time and Evaluation. Time & Society, vol. 8(1), 59–98.

[5] A. Bell, 1997. The Discourse Structure of News Structure. Approaches to Media Discourse, ed. A.Bell, 64–104.

[6] E. Filatova and E. Hovy. 2001. Assigning Time-Stamps to Event-Clauses. Proceedings of ACL Workshop on Temporal and Spatial Reasoning.

[7] M. Lapata, A. Lascarides, 2004, Inferring sentence-internal temporal relations, In Proceedings of the North American Chapter of the Assocation of Computational Linguistics

[8] Nathanael Chambers, Shan Wang, Dan Jurafsky , Classifying Temporal Relations Between Events, ACL-07, Prague. 2007

[9] Inderjeet Mani , Barry Schiffman , Jianping Zhang, Inferring temporal ordering of events in news, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology

[10] Part of Speech Tagging:- http://en.wikipedia.org/wiki/Part-of-speech_tagging

[11]Modality:-http://en.wikipedia.org/wiki/Linguistic_modality

[12]Aspect:- http://en.wikipedia.org/wiki/Grammatical_aspect

[13] Stanford typed dependencies manual:- Marie-Catherine de Marneffe and Christopher D. Manning