Using Query Patterns to Learn the Duration of Events

Andrey Gusev Nathanael Chambers Pranav Khaitan Divye Khilnani
Steven Bethard Dan Jurafsky
Department of Computer Science, Stanford University
{agusev,nc,pranavkh,divyeraj,bethard,jurafsky}@cs.stanford.edu

Abstract

We present the first approach to learning the durations of events without annotated training data, employing web query patterns to infer duration distributions. For example, we learn that "war" lasts *years* or *decades*, while "look" lasts *seconds* or *minutes*. Learning aspectual information is an important goal for computational semantics and duration information may help enable rich document understanding. We first describe and improve a supervised baseline that relies on event duration annotations. We then show how web queries for linguistic patterns can help learn the duration of events without labeled data, producing fine-grained duration judgments that surpass the supervised system. We evaluate on the TimeBank duration corpus, and also investigate how an event's participants (arguments) effect its duration using a corpus collected through Amazon's Mechanical Turk. We make available a new database of events and their duration distributions for use in research involving the temporal and aspectual properties of events.

1 Introduction

Bridging the gap between lexical knowledge and world knowledge is crucial for achieving natural language understanding. For example, knowing whether a nominal is a person or organization and whether a person is male or female substantially improves coreference resolution, even when such knowledge is gathered through noisy unsupervised approaches (Bergsma, 2005; Haghighi and Klein, 2009). However, existing algorithms and resources for such semantic knowledge have focused primarily on static properties of nominals (e.g. gender or entity type), not dynamic properties of verbs and events.

This paper shows how to learn one such property: the typical duration of events. Since an event's duration is highly dependent on context, our algorithm models this aspectual property as a distribution over durations rather than a single mean duration. For example, a "war" typically lasts *years*, sometimes *months*, but almost never *seconds*, while "look" typically lasts *seconds* or *minutes*, but rarely *years* or *decades*. Our approach uses web queries to model an event's typical distribution in the real world.

Learning such rich aspectual properties of events is an important area for computational semantics, and should enrich applications like event coreference (e.g., Chen and Ji, 2009) in much the same way that gender has benefited nominal coreference systems. Event durations are also key to building event timelines and other deeper temporal understandings of a text (Verhagen et al., 2007; Pustejovsky and Verhagen, 2009).

The contributions of this work are:

- Demonstrating how to acquire event duration distributions by querying the web with patterns.
- Showing that a system that predicts event durations based only on our web count distributions can outperform a supervised system that requires manually annotated training data.
- Making available an event duration lexicon with duration distributions for common English events.

We first review previous work and describe our re-implementation and augmentation of the latest supervised system for predicting event durations. Next, we present our approach to learning event distributions based on web counts. We then evaluate both of these models on an existing annotated corpus of event durations and make comparisons to durations we collected using Amazon's Mechanical Turk. Finally, we present a generated database of event durations.

2 Previous Work

Early work on extracting event properties focused on linguistic aspect, for example, automatically distinguishing *culminated* events that have an end point from *non-culminated* events that do not (Siegel and McKeown, 2000). The more fine-grained task of predicting the duration of events was first proposed by Pan et al. (2006), who annotated each event in a small section of the TimeBank (Pustejovsky et al., 2003) with duration lower and upper bounds. They then trained support vector machines on their annotated corpus for two prediction tasks: *less-than-a-day* vs. *more-than-a-day*, and bins like *seconds*, *minutes*, *hours*, etc. Their models used features like bags of words, heads of syntactic subjects and objects, and WordNet hypernyms of the events. This work provides a valuable resource in its annotated corpus and is also a good baseline. We replicate their work and also add new features as described below.

Our approach to the duration problem is inspired by the standard use of web patterns for the acquisition of relational lexical knowledge. Hearst (1998) first observed that a phrase like "... algae, such as Gelidium..." indicates that "Gelidium" is a type of "algae", and so hypernym-hyponym relations can be identified by querying a text collection with patterns like "such <noun> as <noun>" and "<noun> , including <noun>". A wide variety of pattern-based work followed, including the application of the idea in VerbOcean to acquire aspects and temporal structure such as happens-before, using patterns like "to <verb> and then <verb>" (Chklovski and Pantel, 2004).

More recent work has learned nominal gender and animacy by matching patterns like "<noun> * himself" and "<noun> and her" to a corpus of Web n-grams (Bergsma, 2005; Ji and Lin, 2009). Phrases like "John Joseph", which were observed often with masculine pronouns and never with feminine or neuter pronouns, can thus have their gender identified as masculine. Ji and Lin found that such web-counts can predict person names as well as a fully supervised named entity recognition system.

Our goal, then, is to integrate these two strands in the literature, applying pattern/web approaches to the task of estimating event durations. One difference from previous work is the distributional nature of the extracted knowledge. In the time domain, unlike in most previous relation-extraction domains, there is rarely a single correct answer: "war" may last *months*, *years* or *decades*, though *years* is the most likely. Our goal is thus to produce a distribution over durations rather than a single mean duration.

3 Duration Prediction Tasks

In both our supervised and unsupervised models, we consider two types of event duration predictions: a *coarse-grained* task in which we only want to know whether the event lasts more or less than a day, and a *fine-grained* task in which we want to know whether the event lasts *seconds*, *minutes*, *hours*, *days*, *weeks*, *months* or *years*. These two duration prediction tasks were originally suggested by Pan et al. (2006), based on their annotation of a subset of newspaper articles in the Timebank corpus (Pustejovsky et al., 2003). Events were annotated with a minimum and maximum duration like the following:

- **5 minutes – 1 hour**: A Brooklyn woman who was *watching* her clothes dry in a laundromat.
- **1 week – 3 months**: Eileen Collins will be named commander of the Space Shuttle *mission*.
- **3 days – 2 months**: President Clinton says he is *committed* to a possible strike against Iraq...

Pan et al. suggested the *coarse-grained* binary classification task because they found that the mean event durations from their annotations were distributed bimodally across the corpus, roughly split into short events (less than a day) and long events (more than a day). The *fine-grained* classification task provides additional information beyond this simple two way distinction.

For both tasks, we must convert the minimum/maximum duration annotations into single labels. We follow Pan et al. (2006) and take the arithmetic mean of the minimum and maximum durations in seconds. For example, in the first event above, *5 minutes* would be converted into 300 seconds, *1 hour* would be converted into 3600 seconds, the resulting mean would be 1950 seconds, and therefore this event would be labeled *less-than-a-day* for the *coarse-grained* task, and *minutes* for the *fine-grained* task. These labels can then be used directly to train and evaluate our models.

4 Supervised Approach

Before describing our query-based approach, we describe our baseline, a replication and extension of the supervised system from Pan et al. (2006). We first briefly describe their features, which are shared across the coarse and fine-grained tasks, and then suggest new features.

4.1 Pan et. al. Features

The Pan et al. (2006) system included the following features which we also replicate:

Event Properties: The event token, lemma and part of speech (POS) tag.

Bag of Words: The n tokens to the left and right of the event word. However, because Pan et al. found that $n = 0$ performed best, we omit this feature.

Subject and Object: The head word of the syntactic subject and object of the event, along with their lemmas and POS tags. Subjects and objects provide important context. For example, "saw Europe" lasts for *weeks* or *months* while "saw the goal" lasts only *seconds*.

Hypernyms: WordNet hypernyms for the event, its subject and its object. Starting from the first synset of each lemma, three hypernyms were extracted from the WordNet hierarchy. Hypernyms can help cluster similar events together. For example, the event *plan* had three hypernym ancestors as features: *idea*, *content* and *cognition*.

4.2 New Features

We present results for our implementation of the Pan et al. (2006) system in Section 8. However, we also implemented additional features.

Event Attributes: Timebank annotates individual events with four attributes: the event word's tense (past, present, future, none), aspect (e.g., progressive), modality (e.g., *could, would, can,* etc.), and event class (occurrence, aspectual, state, etc.). We use each of these as a feature in our classifier. The aspect and tense of the event, in particular, are well known indicators of the temporal shape of events (Vendler, 1976).

Named Entity Classes: Pan et al. found the subject and object of the events to be useful features, helping to identify the particular sense of the event. We used a named entity recognizer to add more information about the subjects and objects, labeling them as *persons, organizations, locations*, or *other*.

Typed Dependencies: We coded aspects of the subcategorization frame of a predicate, such as transitivity, or the presence of prepositional objects or adverbial modifiers, by adding a binary feature for each typed dependency[1] seen with a verb or noun. We experimented with including the head of the argument itself, but results were best when only the dependency type was included.

Reporting Verbs: Many of the events in Timebank are reporting verbs (*say, report, reply,* etc.). We used a list of reporting verbs to identify these events with a binary feature.

4.3 Classifier

Both the Pan et al. feature set and our extended feature set were used to train supervised classifiers for the two event duration prediction tasks. We experimented with naive bayes, logistic regression, maximum entropy and support vector machine classifiers, but as discussed in Section 8, the maximum entropy model performed best in cross-validations on the training data.

5 Unsupervised Approach

While supervised learning is effective for many NLP tasks, it is sensitive to the amount of available training data. Unfortunately, the training data for event durations is very small, consisting of only 58 news articles (Pan et al., 2006), and labeling further data is quite expensive. This motivates our desire to find an

[1] We parsed the documents into typed dependencies with the Stanford Parser (Klein and Manning, 2003).

approach that does not rely on labeled data, but instead utilizes the large amounts of text available on the Web to search for duration-specific patterns. This section describes our web-based approach to learning event durations.

5.1 Web Query Patterns

Temporal properties of events are often described explicitly in language-specific constructions which can help us infer an event's duration. Consider the following two sentences from our corpus:

- Many *spend hours* surfing the Internet.
- The answer is coming up *in a few minutes*.

These sentences explicitly describe the duration of the events. In the first, the dominating clause *spend hours* tells us how long surfing the Internet lasts (*hours*, not *seconds*), and in the second, the preposition attachment serves a similar role. These examples are very rare in the corpus, but as can be seen, are extremely informative when present. We developed several such informative patterns, and searched the Web to find instances of them being used with our target events.

For each pattern described below, we use Yahoo! to search for the patterns occurring with our events. We collect the total hit counts and use them as indicators of duration. The Yahoo! search API returns two numbers for a query: *totalhits* and *deephits*. The former excludes duplicate pages and limits the number of documents per domain while the latter includes all duplicates. We take the sum of these two numbers as our count (this worked better than either of the two individually on the training data and provides a balance between the benefits of each estimate) and normalize the results as described in Section 5.2. Queries are submitted as complete phrases with quotation marks, so the results only include exact phrase matches. This greatly reduces the number of hits, but results in more precise distributions.

5.1.1 Coarse-Grained Patterns

The coarse grained task is a binary decision: *less than a day* or *more than a day*. We can model this task directly by looking for constructions that can only be used with events that take less than a day. The adverb *yesterday* fills this role nicely; an event modified by *yesterday* strongly implies that it took place within a single day's time. For example, *'shares closed at $18 yesterday'* implies that the *closing* happened in less than a day. We thus consider the following two query patterns:

- <event$_{past}$> yesterday
- <event$_{pastp}$> yesterday

where <event$_{past}$> is the past tense (preterite) form of the event (e.g., *ran*), and <event$_{pastp}$> is the past progressive form of the event (e.g., *was running*).

5.1.2 Fine-Grained Patterns

For the fine-grained task, we need patterns that can identify when an event falls into any of the various buckets: *seconds*, *minutes*, *hours*, etc. Thus, our fine-grained patterns are parameterized both by the event and by the bucket of interest. We use the following patterns inspired in part by Dowty (1979):

1. <event$_{past}$> for * <bucket>
2. <event$_{pastp}$> for * <bucket>
3. spent * <bucket> <event$_{ger}$>

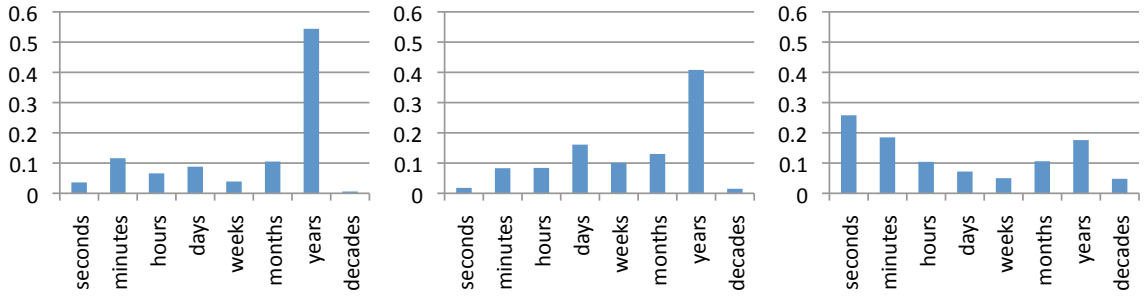
where <event$_{past}$> and <event$_{pastp}$> are defined as above, <event$_{ger}$> is the gerund form of the event (e.g., *running*), and the wildcard '*' can match any single token[2].

The following three patterns ultimately did not improve the system's performance on the training data:

4. <event$_{past}$> in * <bucket>
5. takes * <bucket> to <event>
6. <event$_{past}$> last <bucket>

Pattern 4 returned a lot of hits, but had low precision as it picked up many non-durative expressions. Pattern 5 was very precise but typically returned few hits, and pattern 6 worked for, e.g., *last week*, but did not work for shorter durations. All reported systems use patterns 1-3 and do not include 4-6.

[2]We experimented with varying numbers of wildcards but found little difference in performance on the training data.



(a) "was saying for <bucket>" (b) "for <bucket>" (c) (a) counts divided by (b) counts

Figure 1: Normalizing the distribution for the pattern "was saying for <bucket>".

We also tried adding subjects and/or objects to the patterns when they were present for an event. However, we found that the benefit of the extra context was outweighed by the significantly fewer hits that resulted. We implemented several backoff approaches that removed the subject and object from the query, however, the counts from these backoff approaches were less reliable than just using the base event.

5.2 Predicting Durations from Patterns

To predict the duration of an event from the above patterns, we first insert the event into each pattern template and query the web to see how often the filled template occurs. These counts form a distribution over each of the bins of interest, e.g., in the fine-grained task we have counts for *seconds*, *minutes*, *hours*, etc. We discard pattern distributions with very low total counts, and normalize the remaining pattern distributions based on the frequency with which the pattern occurs in general. Finally, we uniformly merge the distributions from all patterns, and use the resulting distribution to select a duration label for the event. The following sections detail this process.

5.2.1 Coarse-Grained Prediction

For the coarse-grained task of less than a day vs. more than a day, we collect counts using the two *yesterday* patterns described above. We then normalize these counts by the count of the event's occurrence in general. For example, given the event *run*, we query for "ran yesterday" and divide by the count of "ran". This gives us the probability of seeing *yesterday* given that we saw *ran*. We average the probabilities from the two *yesterday* patterns, and classify an event as lasting less than a day if its average probability exceeds a threshold t. We optimized t to our training set ($t = .002$). This basically says that if an event occurs with *yesterday* more than 0.2% of the time, we will assume that the event lasts less than a day.

5.2.2 Fine-Grained Prediction

As with the coarse-grained task, our fine-grained approach begins by collecting counts using the three fine-grained patterns discussed above. Since each fine-grained pattern has both an <event> and a <bucket> slot to be filled, for a single event and a single pattern, we end up making 8 queries to cover each of the 8 buckets: *seconds*, *minutes*, *hours*, *days*, *weeks*, *months*, *years* and *decades*. After these queries, we have a pattern-specific distribution of counts over the various buckets, a coarse measure of the types of durations that might be appropriate to this event. Figure 1(a) shows an example of such a distribution.

As can be seen in Figure 1(a), this initial distribution can be skewed in various ways – in this case, *years* is given far too much mass. This is because in addition to the single event interpretation of words like "saying", there are iterative or habitual interpretations (Moens and Steedman, 1988; Frawley, 1992). Iterative events occur repeatedly over a period of time, e.g., "he's been saying for years that..." The two interpretations are apparent in the raw distributions of *smile* and *run* in Figure 2. The large peak at *years* for *run* shows that it is common to say someone "was running for years." Conversely, it is less common to say someone "was smiling for years," so the distribution for *smile* is less biased towards *years*.

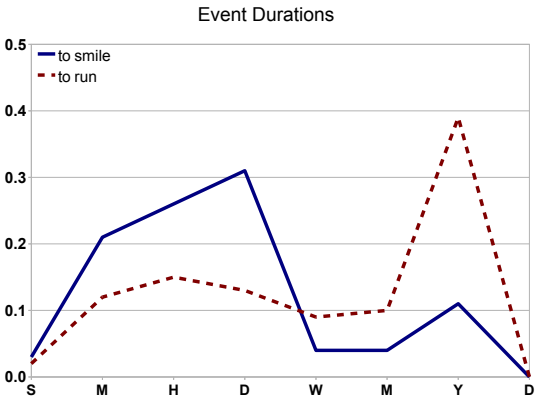


Figure 2: Two double peaked distributions.

Coverage of Fine-Grained Query Patterns

Number of Patterns	Total Events	Precision
At least one	1359 (81.7%)	57.3
At least two	1142 (68.6%)	58.6
All three	428 (25.7%)	65.7

Figure 3: The number of events that match n fine-grained patterns and the pattern precision on these events. The training set consists of 1664 events.

While the problem of distinguishing single events from iterative events is out of the scope of this paper (though an interesting avenue for future research), we can partially address the problem by recognizing that some buckets are simply more frequent in text than others. For example, Figure 1(b) shows that it is by far more common to see "for <bucket>" filled with *years* than with any other duration unit. Thus, for each bucket, we divide the counts collected with the event patterns by the counts we get for the pattern without the event[3]. Essentially, this gives us for each bucket the probability of the event given that bucket. Figure 1(c) shows that the resulting normalized distribution fits our intution of how long "saying" should last much better than the raw counts: *seconds* and *minutes* have much more of the mass now.

After normalizing an event's counts for each pattern, we combine the distributions from the three different patterns if their hit counts pass certain confidence thresholds. The total hit count for each pattern must exceed a minimum threshold $t_{min} = 100$ and not exceed a maximum threshold $t_{max} = 100,000$ (both thresholds were optimized on the training data). The former avoids building distributions from a sparse number of hits, and the latter avoids classifying generic and polysemous events like 'to make' that return a large number of hits. We found such events to produce generic distributions that do not help in classification. If all three patterns pass our confidence thresholds, we merge the pattern distributions by summing them bucket-wise together and renormalizing the resulting distribution to sum to 1. Merging the patterns mitigates the noise from any single pattern.

To predict the event's duration, we then select the bucket with the highest *smoothed* score:

$$score(b_i) = b_{i-1} + b_i + b_{i+1}$$

where b_i is a duration bucket and $0 < i < 9$. We define $b_0 = b_9 = 0$. In other words, the score of the *minute* bucket is the sum of three buckets: *second*, *minute* and *hour*. This parallels the smoothing of the evaluation metric introduced by (Pan et al., 2006) which we also adopt for evaluation in Section 7.

In the case that fewer than three of our patterns matched, we backoff to the majority class (*months* for fine-grained, and *more-than-a-day* for coarse-grained). We experimented with only requiring one or two patterns to match, but found the best results on training when requiring all three. Figure 3 shows the large jump in precision when all three are required. The evaluation is discussed in Section 7.

5.2.3 Coarse-Grained Prediction via Fine-Grained Prediction

We can also use the distributions collected from the fine-grained task to predict coarse-grained labels. We use the above approach and return *less than a day* if the selected fine-grained bucket was *seconds*, *minutes* or *hours*, and *more than a day* otherwise. We also tried summing over the duration buckets: p(seconds) + p(minutes) + p(hours) for *less than day* and p(days) + p(weeks) + p(months) + p(years) + p(decades) for *more than a day*, but the simpler approach outperformed these summations in training.

[3]We also explored normalizing not by the global distribution on the Web, but by the average of the distributions of all the events in our dataset. However, on the training data, using the global distribution performed better.

6 Datasets

6.1 Timebank Duration

As described in Section 3, Pan et al. (2006) labeled 58 documents with event durations. We follow their method of isolating the 10 WSJ articles as a separate test set which we call *TestWSJ* (147 events). For the remaining 48 documents, they split the 2132 event instances into a *Train* and *Test* set with 1705 and 427 events respectively. Their split was conducted over the bag of events, so their train and test sets may include events that came from the same document. Their particular split was unavailable.

We instead use a document-split that divides the two sets into bins of documents. Each document's entire set of events is assigned to either the training set or the test set, so we do not mix events across sets. Since documents often repeat mentions of events, this split is more conservative by not mixing test mentions with the training set. Train, Test, and TestWSJ contain 1664 events (714 unique verbs), 471 events (274 unique), and 147 events (84 unique) respectively. For each base verb, we created queries as described in Section 5.1.2. The train/test split is available at *http://cs.stanford.edu/people/agusev/durations/*.

6.2 Mechanical Turk Dataset

We also collected event durations from Amazon's Mechanical Turk (MTurk), an online marketplace from Amazon where requesters can find workers to solve Human Intelligence Tasks (HITs) for small amounts of money. Prior work has shown that human judgments from MTurk can often be as reliable as trained annotators (Snow et al., 2008) or subjects in controlled lab studies (Munro et al., 2010), particularly when judgments are aggregated over many MTurk workers ("Turkers"). Our motivation for using Turkers is to better analyze system errors. For example, if we give humans an event in isolation (no sentence context), how well can they guess the durations assigned by the Pan et. al. annotators? This measures how big the gap is between a system that looks only at the event, and a system that integrates all available context.

To collect event durations from MTurk, we presented Turkers with an event from the TimeBank (a superset of the events annotated by Pan et al. (2006)) and asked them to decide whether the event was most likely to take *seconds*, *minutes*, *hours*, *days*, *weeks*, *months*, *years* or *decades*. We had events annotated in two different contexts: in isolation, where only the event itself was given (e.g., "allocated"), and in subject-object context, where a minimal phrase including the event and its subject and object was given (e.g., "the mayor allocated funds"). In both types of tasks, we asked 10 Turkers to label each event, and they were paid \$0.0025 for each annotation (\$0.05 for a block of 20 events). To filter out obvious spammers, we added a test item randomly to each block, e.g., adding the event "minutes" and rejecting work from Turkers who labeled this anything other than the duration *minutes*.

The resulting annotations give duration distributions for each of our events. For example, when presented the event "remodeling", 1 Turker responded with *days*, 6 with *weeks*, 2 with *months* and 1 with *years*. These annotations suggest that we generally expect "remodeling" to take weeks, but it may sometimes take more or less. To produce a single fine-grained label from these distributions, we take the duration bin with the largest number of Turker annotations, e.g. for "remodeling", we would produce the label *weeks*. To produce a single coarse-grained label, we use the label *less-than-a-day* if the fine-grained label was *seconds*, *minutes* or *hours* and *more-than-a-day* otherwise.

7 Experiment Setup

As discussed in Section 3, we convert the minimum and maximum duration annotations into labels by converting each to seconds using ISO standards and calculating the arithmetic mean. If the mean is ≤ 86400 seconds, it is considered *less-than-a-day* for the coarse-grained task. The fine-grained buckets are similarly calculated, e.g., X is labeled *days* if $86400 < X \leq 604800$. The Pan et al. (2006) evaluation does not include a *decades* bucket, but our system still uses "decades" in its queries.

We optimized all parameters of both the supervised and unsupervised systems on the training set, only running on test after selecting our best performing model. We compare to the majority class as a baseline,

Coarse-Grained

	Test	TestWSJ
Supervised, Pan	**73.3**	73.5
Supervised, all	73.0	**74.8**

Fine-Grained

	Test	TestWSJ
Supervised, Pan	62.2	61.9
Supervised, all	**62.4**	**66.0**

Figure 4: Accuracies of the supervised maximum entropy classifiers with two different feature sets.

Coarse-Grained

	Test	TestWSJ
Majority class	62.4	57.1
Supervised, all	**73.0***	**74.8***
Web counts, yesterday	70.7*	**74.8***
Web counts, buckets	72.4*	73.5*

Fine-Grained

	Test	TestWSJ
Majority class	59.2	52.4
Supervised, all	62.4	66.0†
Web counts, buckets	**66.5***	**68.7***

Figure 5: System accuracy compared against supervised and majority class. * indicates statistical significance (McNemar's Test, two-tailed) against majority class at the $p < 0.01$ level, † at $p < 0.05$

tagging all events as *more-than-a-day* in the coarse-grained task and *months* in the fine-grained task.

To evaluate our models, we use simple accuracy on the coarse-grained task, and approximate agreement matching as in Pan et al. (2006) on the fine-grained task. In this approximate agreement, a guess is considered correct if it chooses either the gold label or its immediate neighbor (e.g., *hours* is correct if *minutes*, *hours* or *days* is the gold class). Pan et al. use this approach since human labeling agreement is low (44.4%) on the exact agreement fine-grained task.

8 Results

Figure 4 compares the performance of our two supervised models; the reimplementation of Pan et al. (2006) (**Supervised, Pan**), and our improved model with new features (**Supervised, all**). The new model performs similarily to the Pan model on the in-domain **Test** set, but better on the out-of-domain financial news articles in the **TestWSJ** test. On the latter, the new model improves over Pan et al. by 1.3% absolute on the coarse-grained task, and by 4.1% absolute on the fine-grained task. We report results from the maximum entropy model as it slightly outperformed the naive bayes and support vector machine models[4].

We compare these supervised results against our web-based unsupervised systems in Figure 5. For the coarse-grained task, we have two web count systems described in Section 5: one based on the *yesterday* patterns (**Web counts, yesterday**), and one based on first gathering the fine-grained bucket counts and then converting those to coarse-grained labels (**Web counts, buckets**). Generally, these models perform within 1-2% of the supervised model on the coarse-grained task, though the *yesterday*-based classifier exactly matches the supervised system's performance on the TestWSJ data. The supervised system's higher results are not statistically significant against our web-based systems.

For the fine-grained task, Figure 5 compares our web counts algorithm based on duration distributions (Section 5) to the baseline and supervised systems. Our web counts approach outperforms the best supervised system by 4.1% absolute on the Test set and by 2.7% absolute on the out-of-domain TestWSJ.

To get an idea of how much the subject/object context could help predict event duration if integrated perfectly, we evaluated the Mechanical Turk annotations against the Pan et. al. annotated dataset using approximate agreement as described in Section 7. Figure 6 gives the performance of the Turkers given two types of context: just the event itself (**Event only**), and the event plus its subject and/or object (**Event and args**). Turkers performed below the majority class baseline when given only the event, but generally above the baseline when given the subject and object, improving up to 20% over the event-only condition.

Figure 7 shows examples of events with different learned durations.

[4]This differs from Pan et al. who found support vector machines to be the best classifier.

Mechanical Turk Accuracy

	Coarse		Fine	
	Test	WSJ	Test	WSJ
Majority class	62.4	57.1	**59.2**	52.4
Event only	52.0	49.4	42.1	43.8
Event and args	**65.0**	**70.1**	56.7	**59.9**

Figure 6: Accuracy of Mechanical Turkers against Pan et. al. annotations.

Learned Examples	
talk to tourism leaders	minutes
driving	hours
shut down the supply route	days
travel	weeks
the downturn across Asia	months
build a museum	years

Figure 7: Examples of web query durations.

9 Discussion

Our novel approach to learning event durations showed 4.1% and 2.7% absolute gains over a state-of-the-art supervised classifier. Although the gain is not statistically significant, these results nonetheless suggest that we are learning as much about event durations from the web counts as we are currently able to learn with our improvements to Pan et al.'s (2006) supervised system. This is encouraging because it indicates that we may not need extensive manual annotations to acquire event durations. Further, our final query system achieves these results with only the event word, and without considering the subject, object or other types of context.

Despite the fact that we saw little gains in performance when including subjects and objects in our query patterns, the Mechanical Turk evaluation suggests that more information may still be gleaned from the additional context. Giving Turkers the subject and object improved their label accuracy by 10-20% absolute. This suggests that finding a way to include subjects and objects in the web queries, for example by using thesauri to generate related queries, is a valuable line of research for future work.

Finally, these MTurk experiments suggest that classifying events for duration *out of context* is a difficult task. Pan et al. (2006) reported 0.88 annotator agreement on the coarse-grained task when given the entire document context. Out of context, given just the event word, our Turkers only achieved 52% and 49% accuracy. Not surprisingly, the task is more difficult without the document. Our system, however, was also only given the event word, but it was able achieve over 70% in accuracy. This suggests that rich language understanding is often needed to correctly label an event for duration, but in the absence of such understanding, modeling the duration by web counts appears to be a practical and useful alternative.

10 A Database of Event Durations

Given the strong performance of our model on duration classification, we are releasing a database of events and their normalized duration distributions, as predicted by our bucket-based fine-grained model. We extracted the 1000 most frequent verbs from a newspaper corpus (the NYT portion of Gigaword Graff (2002)) with the 10 most frequent grammatical objects of each verb. These $10,000$ events and their duration distributions are available at *http://cs.stanford.edu/people/agusev/durations/*.

Acknowledgements

Thanks to Chris Manning and the anonymous reviewers for insightful comments and feedback. This research draws on data provided by Yahoo!, Inc., through its Yahoo! Search Services offering. We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

Bergsma, S. (2005). Automatic acquisition of gender information for anaphora resolution. In *Advances in Artificial Intelligence*, Volume 3501 of *Lecture Notes in Computer Science*, pp. 342–353. Springer Berlin / Heidelberg.

Chen, Z. and H. Ji (2009). Graph-based event coreference resolution. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, Singapore, pp. 54–57. ACL.

Chklovski, T. and P. Pantel (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In D. Lin and D. Wu (Eds.), *Proceedings of EMNLP 2004*, Barcelona, Spain, pp. 33–40.

Dowty, D. R. (1979). *Word Meaning and Montague Grammar*. Kluwer Academic Publishers.

Frawley, W. (1992). *Linguistic Semantics*. Routledge.

Graff, D. (2002). English Gigaword. *Linguistic Data Consortium*.

Haghighi, A. and D. Klein (2009). Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of EMNLP-2009*, Singapore, pp. 1152–1161.

Hearst, M. A. (1998). Automated discovery of wordnet relations. In *WordNet: An Electronic Lexical Database*. MIT Press.

Ji, H. and D. Lin (2009). Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation*.

Klein, D. and C. D. Manning (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pp. 423–430.

Moens, M. and M. Steedman (1988). Temporal ontology in natural language. *Computational Linguistics 2*(14), 15–21.

Munro, R., S. Bethard, V. Kuperman, V. T. Lai, R. Melnick, C. Potts, T. Schnoebelen, and H. Tily (2010). Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Los Angeles, pp. 122–130.

Pan, F., R. Mulkar, and J. Hobbs (2006). Learning event durations from event descriptions. In *Proceedings of COLING-ACL*.

Pustejovsky, J., P. Hanks, R. Sauri, A. See, D. Day, L. Ferro, R. Gaizauskas, M. Lazo, A. Setzer, and B. Sundheim (2003). The timebank corpus. *Corpus Linguistics*, 647–656.

Pustejovsky, J. and M. Verhagen (2009). Semeval-2010 task 13: Evaluating events, time expressions, and temporal relations (tempeval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, Boulder, Colorado, pp. 112–116.

Siegel, E. V. and K. R. McKeown (2000). Learning methods to combine linguistic indicators: improving aspectual classification and revealing linguistic insights. *Computational Linguistics 26*(4), 595–628.

Snow, R., B. O'Connor, D. Jurafsky, and A. Ng (2008). Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP-2008*, Hawaii.

Vendler, Z. (1976). Verbs and times. *Linguistics in Philosophy*, 97–121.

Verhagen, M., R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky (2007). Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic, pp. 75–80.