



Swansea University
Prifysgol Abertawe



Cronfa - Swansea University Open Access Repository

This is an author produced version of a paper published in:
Research and Development in Intelligent Systems XXX

Cronfa URL for this paper:

<http://cronfa.swan.ac.uk/Record/cronfa43777>

Book chapter :

Blamey, B., Crick, T. & Oatley, G. (2013). *'The First Day of Summer': Parsing Temporal Expressions with Distributed Semantics*. *Research and Development in Intelligent Systems XXX*, (pp. 389-402). Cambridge, UK: Springer.
http://dx.doi.org/10.1007/978-3-319-02621-3_29

This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder.

Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

<http://www.swansea.ac.uk/library/researchsupport/ris-support/>

‘The First Day of Summer’: Parsing Temporal Expressions with Distributed Semantics

Ben Blamey and Tom Crick and Giles Oatley

Abstract Detecting and understanding temporal expressions are key tasks in natural language processing (NLP), and are important for event detection and information retrieval. In the existing approaches, temporal semantics are typically represented as discrete ranges or specific dates, and the task is restricted to text that conforms to this representation. We propose an alternate paradigm: that of *distributed temporal semantics* – where a probability density function models relative probabilities of the various interpretations. We extend SUTime, a state-of-the-art NLP system to incorporate our approach, and build definitions of new and existing temporal expressions. A worked example is used to demonstrate our approach: the estimation of the creation time of photos in online social networks (OSNs), with a brief discussion of how the proposed paradigm relates to the point- and interval- based systems of time. An interactive demonstration, along with source code and datasets, are available online.

1 Introduction

Temporal expressions communicate more than points and intervals on the real axis of unix time – their true meaning is much more complex, intricately linked to our culture, and often difficult to define precisely. Extracting the temporal semantics of text is important in tasks such as event detection [12].

We present a technique for leveraging big-data to capture the *distributed temporal semantics* of various classes of temporal expressions (the term *distributed* began to appear in the context of automatic thesauri construction during the 1990s [8]). Our approach models the inherent ambiguity of traditional temporal expressions, as

Ben Blamey, Tom Crick, Giles Oatley
Cardiff Metropolitan University, Western Avenue, Cardiff CF5 2YB, UK
e-mail: {beblamey, tcrick, goatley}@cardiffmet.ac.uk

well as widening the task to infer semantics from quasi-temporal expressions not previously considered for this task.

In Section 2, we discuss how existing work has overlooked the distributed semantics issue, followed by an outline of our approach in Section 3. In Section 4, we describe a technique for mining a distributed definition from photo metadata downloaded from the Flickr service. Examples are shown in Section 5, with a discussion of cultural nuances we find. In Section 6 the changes to the SUTime framework are described. Section 7 shows an example use of the system, for determining the creation time of Facebook photos – highlighting how the approach facilitates incorporation of a prior probability. Section 8 is a brief discussion of how the approach relates to the point- and interval- based systems of time used in AI, followed conclusions in Section 9, with directions for future work in Section 10.

2 Related Work

Research into temporal expressions has generally focused on their detection and grammatical parsing. Traditionally, systems used hand-coded rules to describe a formal grammar. Popular frameworks using such an approach include: HeidelTime [14], GUTime [11], with the more recent SUTime [6], part of the Stanford CoreNLP framework¹, considered to be a state-of-the-art system, as measured on the TempEval-2 dataset [16].

Consistent with general trends in NLP, more statistical approaches have become popular, where grammars are built through the analysis of large corpora. An example is the development of *grammar of time expressions* [2], to concisely model complex compositional expressions.

Whether hand-coded or machine learnt rules are used, the terminal set of the grammar is generally the months, dates, days of the week, religious festivals, public holidays, usually with an emphasis on the culture of the authors. SUTime can be configured to use the JollyDay² library, which contains definitions of important dates for many cultures – but even in this case the definitions are restricted to the model of discrete intervals. This approach is the most natural and simplest approach to the mathematical modelling of time, used throughout natural science. Work such as that of Allen [1] is often cited as a philosophical underpinning of this model.

Such an approach is useful for describing the physical world with mathematical precision, but is a poor means of describing the cultural definition of temporal language. In cases where it is difficult to assign specific date ranges, the advice is to leave alone:

Some expressions' meanings are understood in some fuzzy sense by the general population and not limited to specific fields of endeavor. However, the general rule is that no VAL is to be specified if they are culturally or historically defined, because there would be a high degree of disagreement over the exact value of VAL. [7, pp. 54]

¹ <http://nlp.stanford.edu/software/corenlp.shtml>

² <http://jollyday.sourceforge.net/index.html>

An advantage of using this restricted, well-defined vocabulary is that it facilitates numerical evaluation of parsing accuracy, and performance can be compared with standard datasets, such as those from the TempEval series [15, 16].

However, this emphasis on grammar has resulted in the research community overlooking the meaning of the terminals themselves. An exciting development is the approach of Brucato et. al. [5], who, noting the maturity of tools developed for the traditional tempex task, widen the scope by to include so-called *named temporal expressions*. First, they created a list of NTEs by parsing tables containing temporal expressions in manually selected Wikipedia articles, merging the results with the JollyDay library. This list is used to train a CRF-based detector, which in turn was used to find completely new NTEs, such as sporting events. However, they note the difficulties of learning definitions for the newly-discovered NTEs:

...it is difficult to automatically learn or infer the link between “New Year’s Day” and 1st January, or the associations between north/south hemisphere and which months fall in summer... [5, pp. 6].

They resort to TIMEX3, a traditional, discrete interval representation. In this paper we present an approach to constructing distributed definitions for temporal expressions, to hopefully overcome this issue.

Clearly, looking for temporal patterns in datasets is not a new task, many studies have observed that Twitter activity relating to some topic or event can peak at or near the corresponding real-world activity. Indeed, work closely related to our approach detects topics through periodicity [18], as compared with existing approaches of Probabilistic Latent Semantic Analysis (PLSA), and Latent Dirichlet Allocation (LDA). Our goal here is to demonstrate how such data can be viewed as a distributed *definition* of the expressions, and that this definition can be incorporated into temporal expression software by choosing a suitable probability density function.

3 A Distributional Approach to Defining Temporal Expressions

We pursue a distributional approach for two reasons: firstly, a distributed definition can capture a more detailed cultural meaning. Examples from our study show that these common temporal expressions are often associated with instances outside their official, or historical definition. We find distributions have a range of skewness and variance, some with more complex patterns exhibiting cultural ambiguity; we discuss specific detailed examples in Section 5.

Secondly, our approach allows a much larger range of expressions to be considered as temporal expressions. Under the current paradigm, phrases need to be associated with specific intervals or instances in time. Religious festivals and public holidays can be resolved to their official meaning, but this is not possible for expressions where no single such definition exists. Indeed, there are many expressions that have consistent temporal meaning, without any universal official dates. See Sec-

tion 5 for a discussion of examples in this category, such as “Freshers’ Week” and “Last Day of School” (Figure 5).

In our theoretical model, we define time $t \in \mathbb{R}$. A temporal expression S is represented by a function $f(t)$, which is a probability density function for the continuous random variable T_s (Section 8 for interpretation of this random variable). For our purposes here, we define a p.d.f. $f(t)$ simply as:

$$P(T_s \geq t_1, T_s \leq t_2) = \int_{t_1}^{t_2} f(t) dt \quad (1)$$

it follows that:

$$\int_{-\infty}^{\infty} f(t) dt = 1 \quad (2)$$

and

$$f(t) \geq 0 \forall t \quad (3)$$

In practice, we work over a smaller, finite date range, suitable for the context. For this paper, we consider a single ‘generic year’, and focus on handling temporal expressions with date-level granularity.

4 Mining the Definitions

Photographs uploaded to the photo-sharing site Flickr³, used in numerous other studies, have been used as the basis for our definitions. The Flickr API was used to search for all photos relating to each term uploaded in the year 2012⁴. Metadata was retrieved for each matching photo⁵, the ‘taken’ attribute of the ‘dates’ element is the photo creation timestamp (Flickr extracts this from the EXIF [3] metadata, if it exists).

Our aim for using an online social network as a data source was to build culturally accurate definitions. A photo-sharing service was used because the semantics of the photo metadata would be more closely associated with the timestamp of the photo than would be the case for a status message. Tweets such as “getting fit for the summer”, “excited about the summer”, “miss the summer”, etc, do not reveal a specific definition of the word or phrase in question. Conversely, a photo labelled “Summer”, “Graduation” or such like, indicates a clear association between the term and the time the photo was taken. Measuring this association on a large scale yields a distributed definition – literally a statistical model of how society defines the term.

For our initial system, we consider only temporal expressions for which we expect the pattern to repeat on an annual basis. To some extent, this obviates some of the error in the photo timestamps, inevitably originating from inaccurate camera

³ <http://www.flickr.com>

⁴ Using the endpoint described at: <http://www.flickr.com/services/api/flickr.photos.search.html>

⁵ Using the endpoint described at: <http://www.flickr.com/services/api/flickr.photos.getInfo.html>

clocks, and timezone issues. A similar approach should be suitable for creating definitions at other scales of time. A range of phrases were chosen for the creation of a distributed definition, some that are commonly used for such purposes, and other more novel examples taken from Facebook photo album titles (Section 5).

Having collected a list of timestamps for each term, we needed to find a probability density function to provide a convenient representation, and smooth the data appropriately. An added complication is that mapping time into the interval of a single year creates an issue when trying to fit, say, a normal distribution to the data. The concentration of probability density may lie very close to one end of the interval (e.g. “New Year’s Eve”), which means we cannot neglect contributions from peaks of probability density that lie in neighbouring years in such cases.

Generally, the timestamps were arranged in distinct clusters, so we computed frequencies for 24-hour intervals, and then attempted to fit mixture models to the data, using the expectation-maximization process, implemented in the Accord.NET scientific computing framework [13]. Initial attempts used a mixture of von Moses distributions, a close approximation to the *wrapped normal distribution*, the result of wrapping the normal distribution around the unit circle. We had difficulties reaching a satisfactory fit with this model, so instead we used a mixture of normal distributions, adapted to work under modulo arithmetic (using the so-called *mean of circular quantities*). Hence, the probability density greater than ± 6 months away from the mean is neglected, for each normal distribution in the mixture. With standard deviations typically in the region of a few days, this is reasonable.

After mixed results using k -means clustering to initialize the model, we settled on a uniform arrangement of normal distributions. A uniform distribution was also included to model the background activity level – without this, because the normal distributions had standard deviations of just a few days, fitting was disrupted by the presence of many outliers.

After fitting, normal distributions with a mixing coefficient of less than 0.001 are pruned from the model (as our primary goal is to generate a terse semantic representation). As discussed in Section 10, the inclusion of asymmetric distributions in the mixture, to better model some of the distributions in the data is an obvious avenue for future investigation.

5 Discussion of Fitted Distributions

For **Bonfire Night** (Figure 1) (5th November, United Kingdom) the primary concentration is near the primary date, but with more variance than is with the case with **April Fools’ Day**. A number of other distributions in the fit have between 1-2% mixing coefficient, with means at 8th January (possibly relating to the solemnity of John the Baptist on 16th January), 26th June (Midsummer’s Eve, 23rd June is popular for bonfires in Ireland), and 2nd May (Bonfires are popular in Slavic Europe on 1st May).

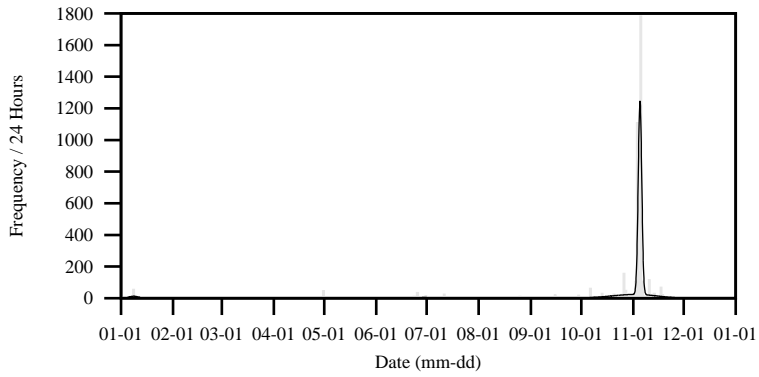


Fig. 1 Distribution of “Bonfire Night”

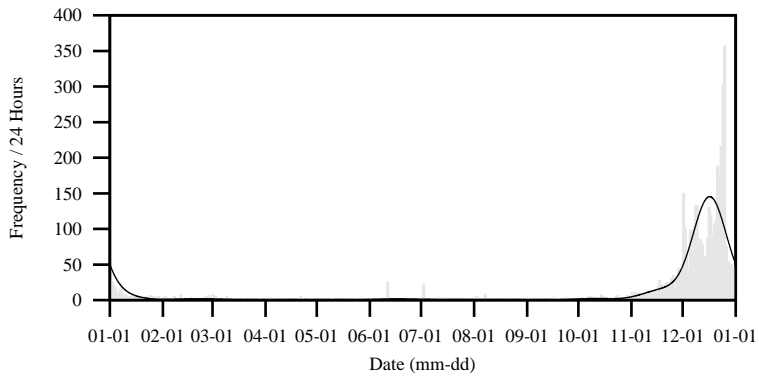


Fig. 2 Distribution of “Christmas”

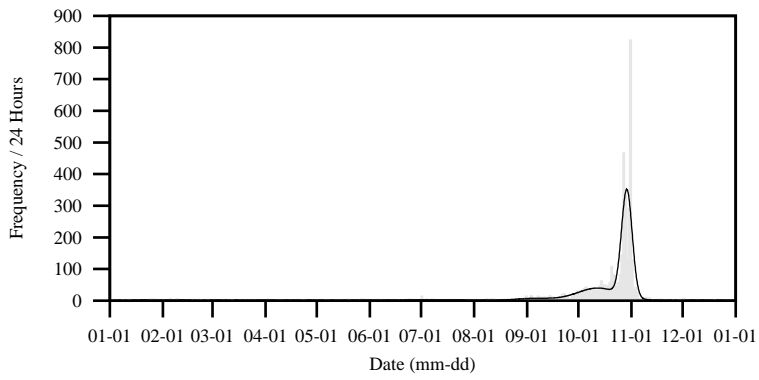


Fig. 3 Distribution of “Halloween”

'The First Day of Summer': Parsing Temporal Expressions with Distributed Semantics

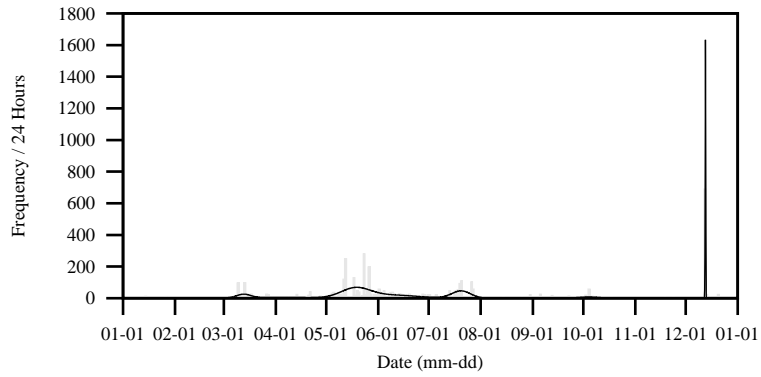


Fig. 4 Distribution of "Graduation"

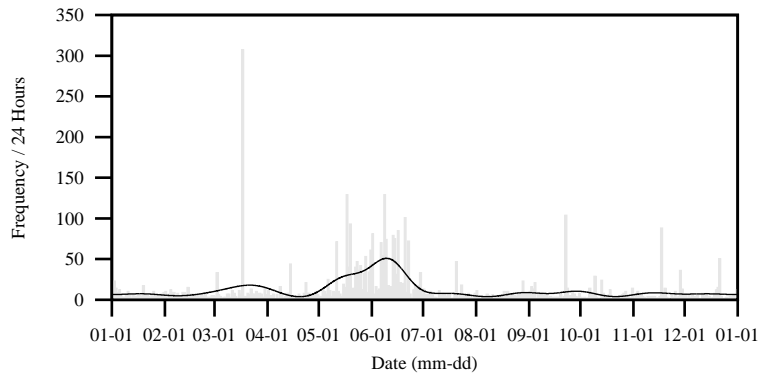


Fig. 5 Distribution of "Last Day of School"

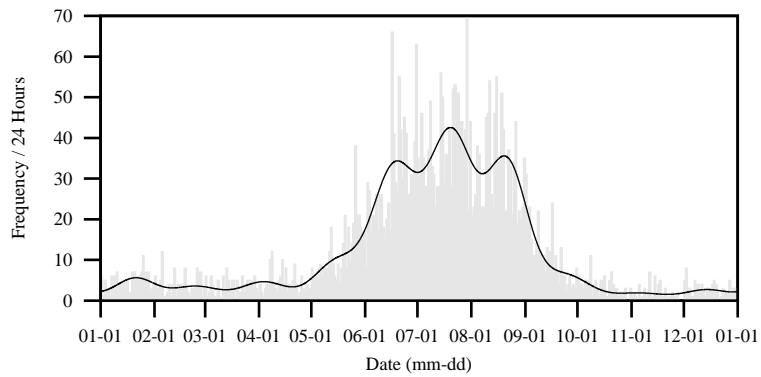


Fig. 6 Distribution of "Summer"

Christmas (Figure 2) (commonly 25th December) starts early, with 10% of the probability density contributed by a distribution with a mean of 20th November. In cultures using the Julian calendar, Christmas is celebrated on 7th January and 19th January, perhaps explaining some of the probability density we see in January. In the case of **New Years' Eve**, more than 92% of the probability mass is centred around 31st of December. We find a normal with mean of January 15th, possibly relating to the Chinese New Year on (23rd January, 2012).

The data for **Halloween** (Figure 3) clearly has a skewed distribution about its official date, 31st October – we see much more activity in the preceding weeks, with activity rapidly dropping off afterwards. A similar distribution is exhibited in the case of **Valentines' Day** on (14th February). A limitation of our work is that we did not include asymmetric distributions in our mixture model; such distributions are fitted to a cluster of normal distributions with appropriately decaying mixture coefficients.

Freshers' Week is a term used predominantly in the UK to describe undergraduate initiation at university, usually in September or October. With the obvious differences between educational calendars between regions and institutions, a complex pattern is unsurprising. In the case of **Last Day of School** (Figure 5), assuming a precise date range in the general case is clearly impossible. Many universities have multiple **Graduation** (Figure 4) ceremonies a year, with loose conventions on dates, reflected in the clustering of the data.

Definitions of seasons show a significant bias toward northern hemisphere definitions, to be expected with the bias towards English language. However, we do see density at the antipodal dates in each case, modelled by normal distributions of appropriate means. For example, **Winter** has a distribution with a mean of 13th July, with a mix coefficient of 1%, with a similar phenomenon in the other samples. It is clear from the distribution of the data that all season terms we studied are used year-round. The data for **Summer** is shown in Figure 6, exhibiting a similar antipodal peak.

Some degree of background “noise” was present in many of the examples: the mixing coefficient was typically in the region of 2%.

6 Modifications to SUTime

When modifying the SUTime framework [6], our aim was to preserve the existing functionality, as well as implement our distributed approach. A number of Java classes are used to represent the parsed temporal information, our key modification was to augment these classes so that they stored a representation of a probability distribution alongside their other fields. Modifications were then made to the grammar definitions to ensure that instances of these classes were associated with the appropriate probability distributions upon creation, and updated appropriately during grammatical composition.

In more detail, we begin with the core temporal classes. Where appropriate, we added a class field which could optionally hold an object representing the associated probability distribution. Effectively, this object is a tree whose nodes are instances of various new classes: `AnnualNormalDistribution`, `AnnualUniformDistribution`, (as the leaves of the tree), and those representing either a `Sum` or `Intersection` (i.e. multiplication) as the internal nodes. When no distributed definition was available (e.g. when parsing “2012”), a representation of the appropriate discrete interval is used as a leaf⁶. These classes implemented a method to return an expression string suitable for use in `gnuplot` (visible in the online demo⁷) – with the two internal nodes algebraically composing the expressions returned by their children in the obvious way. Generation of an alternative syntax, or support for numerical integration could be implemented as additional methods. We also introduced a new temporal class, to represent a temporal expression which does not have a non-distributed definition (such as “Last Day of School”), for which composition is possible under the distributed paradigm, but which uses a dummy implementation under the traditional paradigm.

Secondly, it was necessary to make a number of changes to the grammar definitions – these files control how instances of temporal classes are created from the input text, and also how the instances of these classes are combined and manipulated based on the underlying text. After fitting the mixture models to the Flickr data (Section 4), definitions were generated in the syntax used by `SUTime`. Rules defining the initial detection of these expressions were updated so that the probability distributions were included, and modified to allow misspellings and repeated characters that we found in Facebook photo album names (common to online social networks [4]). We introduced rules to detect our new temporal expressions, and assign their distributed definitions. Other modifications were made to adapt the grammar to our domain of photo album names, relating to British English date conventions, and to support temporal expressions of the form ‘YY’. The rules for temporal composition were largely unchanged, as they are expressed in terms of the temporal operators defined separately.

`SUTime` defines 17 algebraic operators for temporal instances (e.g. `THIS`, `NEXT`, `UNION`, `INTERSECT`, `IN`). Facebook photo album names tended to contain mostly absolute temporal expressions (none of the form “2 months”, or “next week”), and it was only necessary to modify the `INTERSECT` operator. In the distributed paradigm, intersecting two temporal expressions such as “Xmas” and “2012” is simply a case of multiplying their respective probability density functions. The existing implementation of the operator is unaffected. Adaptation of ‘discrete’ operators such as `PREV` and `NEXT` into the distributed paradigm presents an interesting problem, and is left for future work. All that remained was to include an expression for the final probability density functions in the `TimeML` output⁸.

⁶ `tm_year(x)` and related `gnuplot` functions were useful for this [17, pp. 27].

⁷ <http://benblamey.name/tempex>

⁸ We introduced an ‘X-GNUPlot-Function’ attribute on the `TIMEX3` element for this purpose

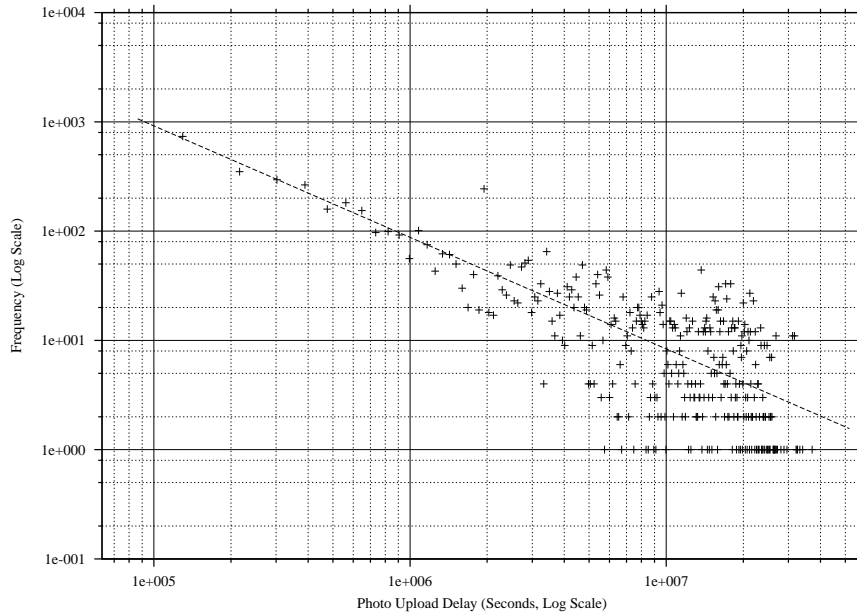


Fig. 7 Distribution of the *Upload Delay*, estimated from Flickr photo metadata.

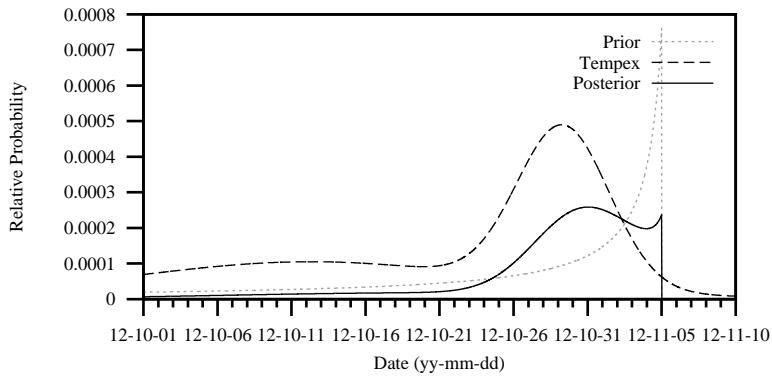


Fig. 8 Computation of the posterior probability distribution for the creation time of the photo, from the prior probability, and the distribution associated the temporal expression “Halloweeeeennn!”.

7 A Worked Example

In tasks such as event detection, it is useful to know the time that a photograph was taken. In Facebook, the EXIF metadata is removed for privacy-related reasons [9], and the API does not publish the photo creation time (as is the case with Flickr). In

Facebook, photo album titles tend to be rich in temporal expressions, and an album title such as “Halloweeeeennnn!” should indicate the date the photo was originally taken, even if it wasn’t uploaded to Facebook until later. The usual technique would be to parse the temporal expression and resolve it to its ‘official’ meaning – in this case, October 31st.

In Figure 3, we see that some of the probability density for “Halloween” actually lies before this date – peaking around the 29th (although the effect is greater with “Christmas”, Figure 2). Having represented the temporal expression as a probability density function, we can combine it with a prior probability distribution, computed as follows. The photo metadata collected from Flickr (Section 4), contains an upload timestamp⁹ in addition to the photo creation timestamp. We define the *upload delay* to be the time difference between the user taking the photo, and uploading it to the web. Figure 7 shows the distribution (tabulated into frequencies for 24-hour bins), plotted with a log-log scale. Taking y as the frequency, and x as the upload delay in seconds, the line of best fit was computed (with gnuplot’s implementation of the Levenberg-Marquardt algorithm) as:

$$\log(y) = a \log(x) + b \quad (4)$$

with:

$$a = -1.0204 \quad (5)$$

$$b = 18.5702 \quad (6)$$

We can then use this equation as a prior distribution for the *creation* timestamp of the photo, by working backwards from the *upload* timestamp, which is available from Facebook¹⁰. Figure 8 shows this prior probability, the distribution for “Halloweeeeennnn!”, and the prior distribution obtained by multiplying them together, respectively scaled for clarity. The resulting posterior distribution has much greater variance than what would have resulted from simply parsing the official definition of October 31st, accounting for events being held on the surrounding days, whilst the application of the prior probability has resulted in a cut-off and a much thinner tail for earlier in the month.

8 Interpreting $S \sim T_s(t)$

In Section 3, we discussed the association (denoted by \sim) between the temporal expression S and the continuous random variable $T_s(t)$. Detailed discussion is beyond the scope of this work, but we briefly outline a few interpretations:

1. S represents some unknown instant: $S \sim t_s$. $T_s(t)$ models $P(t = t_s)$.

⁹ The time when the photo was uploaded to the web, shown as the ‘posted’ attribute of the ‘dates’ element, see: <http://www.flickr.com/services/api/flickr.photos.getInfo.html>

¹⁰ See the ‘created_time’ field at: <https://developers.facebook.com/docs/reference/api/photo/>

2. S represents some unknown interval: $S \sim I_S = [t_a, t_b]$. $T_S(t)$ models $P(t \in I_S)$.
3. The meaning of S is precisely $S \sim T_S(t)$, and only by combining T_S with additional information can anything further be inferred. Particular instances or time intervals may have cultural or historical associations with S , it may be possible to recognize their effect on T_S . But T_S itself is the pragmatic interpretation of S .

There is extensive discussion relating to the models underlying (1) and (2) in the literature, and one can construct various thought experiments to create paradoxes in either paradigm. By constructing our probability distribution by modelling time as \mathbb{R} which means we are undeniably using the classical point-based model of time, rather than the interval-based model [1]. However, the situation is a little more subtle: employing a probability density function only allows computation of the probability associated with an arbitrary *interval* (see Equation 1). For a continuous random variable, the probability of any particular instance is zero by definition; something which is arguably more akin to an interval-based interpretation of time. So, associating the temporal expression with a p.d.f. means that the theoretical basis of the point-based system of time is retained, whilst the mathematics restricts us to working only with intervals. Whether this dual-nature obviates the dividing instant problem [10], requires a more rigorous argument, is something we leave to future work.

The thrust of our contribution is to suggest that temporal expressions in isolation are intrinsically ambiguous¹¹ (interpretation (3)). We argue that such expressions cannot be resolved to discrete intervals or instants (without loss of information), and attempts to do so are perhaps unnecessary or misguided. In some cases, it may be desirable to defer resolution, perhaps to apply a prior probability (as in Section 7).

9 Conclusions

In Section 2 we have discussed how existing work focuses grammar and composition. Recent work to widen the task [5] requires methods (such as ours) for assigning meaning to these expressions. We’ve noted how the usual approach of representing meaning as discrete intervals limits the scope of the temporal expression task.

Our main contribution is a proposal for an alternative *distributed* paradigm for parsing temporal expressions (Section 3). The approach has several advantages:

- It is able to provide definitions for a wider class of temporal expressions, supporting expressions where there is no single official definition.
- It captures greater cultural richness and ambiguity – arguably a more accurate definition.
- It facilitates further processing, such as the consideration of a prior probability, as demonstrated with an example in Section 7.

¹¹ The “weekend”, and precisely when it starts, is a good example of this. Readers will be able to imagine many different possible interpretations of the word.

9. James, J.: How facebook handles image exif data (2011). URL <http://windowsitpro.com/blog/how-facebook-handles-image-exif-data>
10. Ma, J., Knight, B.: Representing the dividing instant. *Comput. J.* **46**(2), 213–222 (2003). URL <http://dblp.uni-trier.de/db/journals/cj/cj46.html#MaK03>
11. Mani, I., Wilson, G.: Robust temporal processing of news. In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 69–76 (2000)
12. Ritter, A., Mausam, Etzioni, O., Clark, S.: Open domain event extraction from twitter. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12*, pp. 1104–1112. ACM, New York, NY, USA (2012). DOI 10.1145/2339530.2339704. URL <http://doi.acm.org/10.1145/2339530.2339704>
13. Souza, C.R.: The Accord.NET Framework (2012). URL <http://accord.googlecode.com>
14. Strötgen, J., Gertz, M.: HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In: *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval'10)*, pp. 321–324 (2010)
15. UzZaman, N., Llorens, H., Allen, J.F., Derczynski, L., Verhagen, M., Pustejovsky, J.: TempEval-3: Evaluating events, time expressions, and temporal relations. In: *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval'13)*, pp. 1–9 (2013)
16. Verhagen, M., Pustejovsky, J.: Temporal processing with the TARSQI toolkit. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 189–192 (2008)
17. Williams, T., Kelley, C.: gnuplot 4.6 (2013). URL <http://www.gnuplot.info/documentation.html>
18. Yin, Z., Cao, L., Han, J., Zhai, C., Huang, T.: LPTA: A probabilistic model for latent periodic topic analysis. In: *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM'11)*, pp. 904–913. IEEE Computer Society (2011)