Temporal Classification of Text and Automatic Document Dating

Angelo Dalli
University of Sheffield
211, Portobello Street
Sheffield, S1 4DP, UK
angelo@dcs.shef.ac.uk

Abstract

Temporal information is presently under-utilised for document and text processing purposes. This work presents an unsupervised method of extracting periodicity information from text, enabling time series creation and filtering to be used in the creation of sophisticated language models that can discern between repetitive trends and non-repetitive writing pat-terns. The algorithm performs in O(n log n) time for input of length n. The temporal language model is used to create rules based on temporal-word associations inferred from the time series. The rules are used to automatically guess at likely document creation dates, based on the assumption that natural languages have unique signatures of changing word distributions over time. Experimental results on news items spanning a nine year period show that the proposed method and algorithms are accurate in discovering periodicity patterns and in dating documents automatically solely from their content.

1 Introduction

Various features have been used to classify and predict the characteristics of text and related text documents, ranging from simple word count models to sophisticated clustering and Bayesian models that can handle both linear and non-linear classes. The general goal of most classification research is to assign objects from a pre-defined domain (such as words or entire documents) to two or more classes/categories. Current and past research has largely focused on solving problems like tagging, sense disambiguation, sentiment classification, author and language identification and topic classification. We introduce an unsupervised method that classifies text and documents according to their predicted time of writing/creation. The method uses a sophisticated temporal language model to predict likely creation dates for a document, hence dating it automatically. This short paper presents some background information about existing techniques and the implemented system, followed by a brief explanation of the classification and dating method, and finally concluding with results and evaluation performed on the LDC GigaWord English Corpus (LDC, 2003).

2 Background

Temporal information is presently under-utilised for document and text processing purposes. Past and ongoing research work has largely focused on the identification and tagging of temporal expressions, with the creation of tagging methodologies such as TimeML/TIMEX (Gaizauskas and Setzer, 2002; Pustejovsky et al., 2003; Ferro et al., 2004), TDRL (Aramburu and Berlanga, 1998) and associated evaluations such as the ACE TERN competition (Sundheim et al. 2004).

Temporal analysis has also been applied in Question-Answering systems (Pustejovsky et al., 2004; Schilder and Habel, 2003; Prager et al., 2003), email classification (Kiritchenko et al.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 29–32,
New York, June 2006.

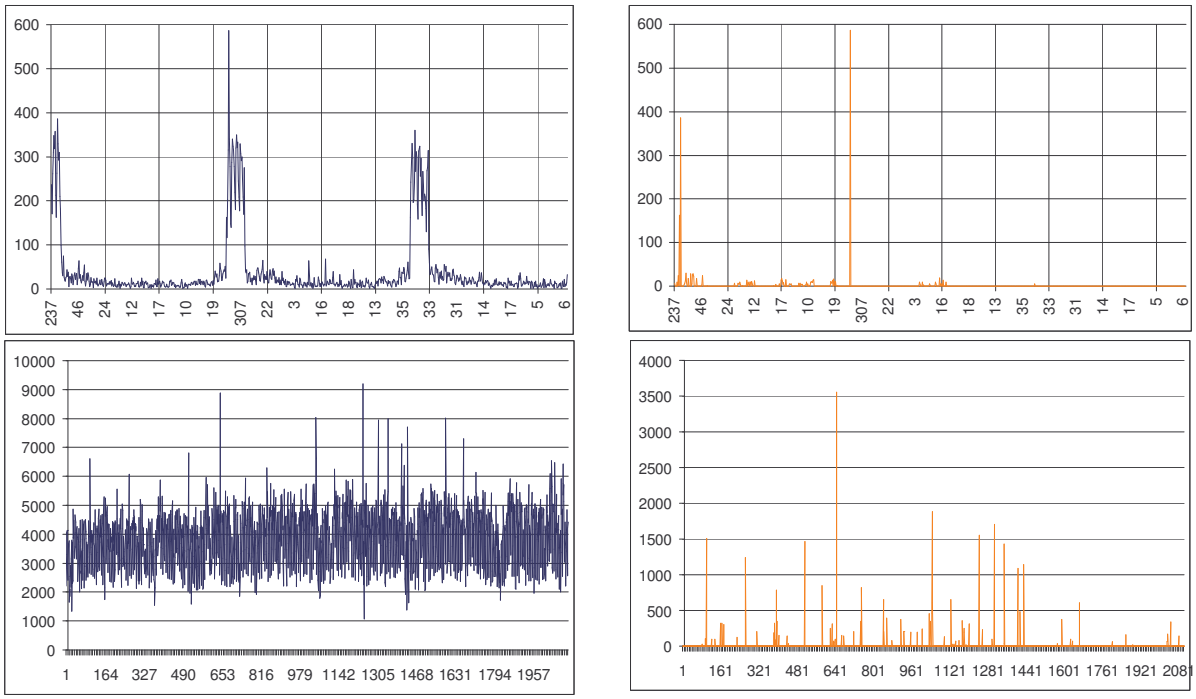

Figure 1 Effects of applying the temporal periodical algorithm on time series for "January" (top) and "the" (bottom) with original series on the left and the remaining time series component after filtering on the right. Y-axis shows frequency count and X-axis shows the day number (time).

2004), aiding the precision of Information Retrieval results (Berlanga et al., 2001), document summarisation (Mani and Wilson, 2000), time stamping of event clauses (Filatova and Hovy, 2001), temporal ordering of events (Mani et al., 2003) and temporal reasoning from text (Boguraev and Ando, 2005; Moldovan et al., 2005). There is also a large body of work on time series analysis and temporal logic in Physics, Economics and Mathematics, providing important techniques and general background information. In particular, this work uses techniques adapted from Seasonal Auto-Regressive Integrated Moving Average models (SARIMA). SARIMA models are a class of seasonal, non-stationary temporal models based on the ARIMA process (defined as a non-stationary extension of the stationary ARMA model). Non-stationary ARIMA processes are defined by:

$$(1-B)^d \phi(B) X_t = \theta(B) Z_t \qquad (1)$$

where d is non-negative integer, and $\phi(X)$ $\theta(X)$ polynomials of degrees p and q respectively. The exact parameters for each process (one process per word) are determined automatically by the system. A discussion of the general SARIMA model is beyond the scope of this paper (details can be found in Mathematics & Physics publications). The NLP application of temporal classification and prediction to guess at likely document and text creation dates is a novel application that has not been considered much before, if at all.

3 Temporal Periodicity Analysis

We have created a high-performance system that decomposes time series into two parts: a periodic component that repeats itself in a predictable manner, and a non-periodic component that is left after the periodic component has been filtered out from the original time series. Figure 1 shows an example of the filtering results on time-series of the words “January” and “the”. The time series are based on training documents selected at random from the GigaWord English corpus. 10% of all the documents in the corpus were used as training documents, with the rest being available for evaluation and testing. A total of 395,944 time series spanning 9 years were calculated from the GigaWord corpus. Figure 2 presents pseudo-code for the time series decomposition algorithm:

1. Find min/max/mean and standard deviation of time series
2. Start with a pre-defined maximum window size (presently set to 366 days)
3. While window size bigger than 1 repeat steps a. to d. below:
    a. Look at current value in time series (starting first value)
    b. Do values at positions current, current + window size, current + 2 x window size, etc. vary by less than ½ standard deviation?
    c. If yes, mark current value/window size pair as being possible decomposition match
    d. Look at next value in time series until the end is reached
    e. Decrease window size by one
4. Select the minimum number of decomposition matches that cover the entire time series using a greedy algorithm

Figure 2 Time Series Decomposition Algorithm

The time series decomposition algorithm was applied to the 395,944 time series, taking an average of 419ms per series. The algorithm runs in $O(n \log n)$ time for a time series of length n.

The periodic component of the time series is then analysed to extract temporal association rules between words and different "seasons", including Day of Week, Week Number, Month Number, Quarter, and Year. The procedure of determining if a word, for example, is predominantly peaking on a weekly basis, is to apply a sliding window of size 7 (in the case of weekly periods) and determining if the periodic time series always spikes within this window. Figure 3 shows the frequency distribution of the periodic time series component of the days of week names ("Monday", "Tuesday", etc.) Note that the frequency counts peak exactly on that particular day of the week. For example, the word "Monday" is automatically associated with Day 1, and "April" associated with Month 4. The creation of temporal association rules generalises inferences obtained from the periodic data. Each association rule has the following information:

Word ID
Period Type (Week, Month, etc.)
Period Number and Score Matrix

The period number and score matrix represent a probability density function that shows the likelihood of a word appearing on a particular period number. For example, the score matrix for "January" will have a high score for period 1 (and period type set to Monthly). Figure 4 shows some examples of extracted association rules. The PDF scores are shown in Figure 4 as they are stored internally (as multiples of the standard deviation of that time series) and are automatically normalised during the classification process at runtime. Rule generalisation is not possible in such a straightforward manner for the non-periodic data. The use of non-periodic data to optimise the results of the temporal classification and automatic dating system is not covered in this paper.

4 Temporal Classification and Dating

The periodic temporal association rules are utilised to automatically guess at the creation date of documents automatically. Documents are input into the system and the probability density functions for each word are weighted and added up. Each PDF is weighted according to the inverse document frequency (IDF) of each associated word. Periods that obtain high score are then ranked for each type of period and two guesses per period type are obtained for each document. Ten guesses in total are thus obtained for Day of Week, Week Number, Month Number, Quarter, and Year (5 period types x 2 guesses each).

	Su	M	T	W	Th	F	S
0	**22660**	10540	7557	772	2130	3264	11672
1	12461	**37522**	10335	6599	1649	3222	3414
2	3394	18289	**38320**	9352	7300	2543	2261
3	2668	4119	18120	**36933**	10427	5762	2147
4	2052	2602	3910	17492	**36094**	9098	5667
5	5742	1889	2481	2568	17002	**32597**	7849
6	7994	7072	1924	1428	3050	14087	**21468**
Av	8138	11719	11806	10734	11093	10081	7782
St	7357	12711	12974	12933	12308	10746	6930

Figure 3 Days of Week Temporal Frequency Distribution for extracted Periodic Component displayed in a Weekly Period Type format

January
Week *1* *2* *3* *4* *5*
Score 1.48 2.20 3.60 3.43 3.52
Month *1* Score 2.95
Quarter *1* Score 1.50

Christmas
Week *2* *5* *36* *42* *44*
Score 1.32 0.73 1.60 0.83 1.32

```
Week   47     49     50     51     52
Score  1.32   2.20   2.52   2.13   1.16

Month  1      9      10     11     12
Score  1.10   0.75   1.63   1.73   1.98
Quarter       4      Score 1.07
```

Figure 4 Temporal Classification Rules for Periodic Components of "January" and "Christmas"

5 Evaluation, Results and Conclusion

The system was trained using 67,000 news items selected randomly from the GigaWord corpus. The evaluation took place on 678,924 news items extracted from items marked as being of type "story" or "multi". Table 1 presents a summary of results. Processing took around 2.33ms per item.

Type	Correct	Incorrect	Avg. Error
DOW	218,899 (32.24%)	460,025 (67.75%)	1.89 days
Week	24,660 (3.53%)	654,264 (96.36%)	14.37 wks
Month	122,777 (18.08%)	556,147 (81.91%)	2.57 mths
Quarter	337,384 (49.69%)	341,540 (50.30%)	1.48 qts
Year	596,009 (87.78%)	82,915 (12.21%)	1.74 yrs
Combined	**422,358 (62.21%)**	**256,566 (37.79%)**	**210 days**

Table 1 Evaluation Results Summary

The actual date was extracted from each news item in the GigaWord corpus and the day of week (DOW), week number and quarter calculated from the actual date. Average errors for each type of classifier were calculated automatically. For results to be considered correct, the system had to have the predicted value ranked in the first position equal to the actual value (of the type of period). The system results show that reasonable accurate dates can be guessed at the quarterly and yearly levels. The weekly classifier had the worst performance of all classifiers. The combined classifier uses a simple weighted formula to guess the final document date using input from all classifiers. The weights for the combined classifier have been set on the basis of this evaluation. The temporal classification and analysis system presented in this paper can handle any Indo-European language in its present form. Further work is being carried out to extend the system to Chinese and Arabic. Current research is aiming at improving the accuracy of the classifier by using the non-periodic components and improving the combined classification method.

References

Aramburu, M. Berlanga, R. 1998. *A Retrieval Language for Historical Documents*. LNCS, 1460, pp. 216-225.

Berlanga, R. Perez, J. Aramburu, M. Llido, D. 2001. *Techniques and Tools for the Temporal Analysis of Retrieved Information*. LNCS, 2113, pp. 72-81.

Boguraev, B. Ando, R.K. 2005. *TimeML-Compliant Text Analysis for Temporal Reasoning*. IJCAI-2005.

Ferro, L. Gerber, L. Mani, I. Sundheim, B. Wilson, G. 2004. *TIDES Standard for the Annotation of Temporal Expressions*. The MITRE Corporation.

Filatova, E. Hovy, E. 2001. *Assigning time-stamps to event-clauses*. Proc. EACL 2001, Toulouse, France.

Gaizauskas, R. Setzer, A. 2002. *Annotation Standards for Temporal Information in NL*. Proc. LREC 2002.

Kiritchenko, S. Matwin, S. Abu-Hakima, S. 2004. *Email Classification with Temporal Features*. Proc. IIPWM 2004, Zakopane, Poland. pp. 523-534.

Linguistic Data Consortium (LDC). 2003. English Gigaword Corpus. David Graff, ed. LDC2003T05.

Mani, I. Wilson, G. 2000. *Robust temporal processing of news*. Proc. ACL 2000, Hong Kong.

Mani, I. Schiffman, B. Zhang, J. 2003. *Inferring temporal ordering of events in news*. HLT-NAACL 2003.

Moldovan, D. Clark, C. Harabagiu, S. 2005. *Temporal Context Representation and Reasoning*. IJCAI-2005.

Prager, J. Chu-Carroll, J. Brown, E. Czuba, C. 2003. *Question Answering using predictive annotation*.

Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, R. Setzer, A. Katz, G. 2003. *TimeML: Robust Specification of event and temporal expressions in text*. IWCS-5.

Pustejovsky, J. Sauri, R. Castano, J. Radev, D. Gaizauskas, R. Setzer, A. Sundheim, B. Katz, G. 2004. "Representing Temporal and Event Knowledge for QA Systems". *New Directions in QA*, MIT Press.

Schilder, F. Habel, C. 2003. *Temporal Information Extraction for Temporal QA*. AAAI NDQA, pp. 35-44.

Sundheim, B. Gerber, L. Ferro, L. Mani, I. Wilson, G. 2004. *Time Expression Recognition and Normalization (TERN)*. http://timex2.mitre.org.