Interactive System for Reasoning about Document Age

Adam Jatowt[1] and Ricardo Campos[2]

[1]Kyoto University
Yoshida-Honmachi, Sakyo-ku
606-8501 Kyoto, Japan
adam@dl.kuis.kyoto-u.ac.jp

[2]Polytechnic Institute of Tomar
LIAAD – INESC TEC
Portugal
ricardo.campos@ipt.pt

ABSTRACT

Recently, many historical texts have become digitized and made accessible for search and browsing. Professionals who work with collections of such texts often need to verify the correctness of documents' key metadata - their creation dates. In this paper, we demonstrate an interactive system for estimating the age of documents. It may be useful not only for tagging a large number of undated documents, but also for verifying already known timestamps. In order to infer probable dates, we rely on a large scale lexical corpora, *Google Books Ngrams*. Besides estimating the document creation year, the system also outputs evidences to support age detection and reasoning process and allows testing different hypotheses about document's age.

Keywords

Document metadata, historical texts, document timestamping

1. INTRODUCTION

In recent years we have witnessed massive digitalization of historical texts carried by libraries, museums and other memory institutions. Old books, news articles, letters and other documents have been scanned, subject to optical character recognition and made available to public. Project Gutenberg[1], Google Books[2] and Internet Archive Text Collection[3] are examples of such initiatives. Many times, custodians of such collections or researchers do not know the exact age of archived texts, or they may be suspicious about the correctness of document creation dates. Given an input document, a professional may then wish to know whether the provided timestamp is accurate or, in cases when the timestamp is missing, to infer it by automatically "carbon-dating" the document content. Automatically generating or verifying temporal metadata of historical texts should help with document annotation, management, authorship attribution and, in general, with their better understanding. Typical approaches to document timestamping are based on the phenomenon of the language change over time [8] and rely on employing features derived from temporal language models, diachronic frequencies of words or occurrences of named entities, and etc. Popescu and Strapparava [10] for example, have organized the Diachronic Text Evaluation challenge under the umbrella of SemEval workshop series to foster the development of algorithms capable of identifying the time interval a given document was published in. A number of research articles have also been published based on heuristic methods [3,6,7], language models [2,4,5] or utilizing diachronic frequencies of words [1,11] to determine document age. Despite these advances, still there is lack of a system for computing document age in an interactive way. Furthermore, existing solutions only generate the final answers and, in general, they do not output convincing evidences to support calculated document creation dates. We believe that interactive visualization systems are needed for professionals working with documents who should appreciate investigatory functionalities to test diverse hypotheses and to formulate their final judgments based on the sets of concrete proofs.

The proposed demo provides, in a visual and interactive way, contextual temporal knowledge about input documents based on the associated large scale and long-term corpora. The system is available online[4] to experiment with. Besides directly helping to reason about the document date, it can be useful in supporting the design of more complex methods for timestamping historical documents, it could be potentially helpful for improving OCR error recognition and can support general document investigation.

2. DATASET

To accomplish our objectives, we need a dataset large enough to support drawing valid conclusions. We use *Google Books Ngrams*[5] compiled from the Google Books project which claims to contain data derived from about 5% of books ever published. The datasets were created in 2009 based on automatically scanned books which were originally published between 1600 and 2000 and were subject to Optical Character Recognition (OCR). The data on ngram frequency is available for each year for the last two centuries. For example, on average, 1-gram dataset contains 17.9 billion words per decade. This demands efficient infrastructure to store and effectively utilize the whole data.

Google Books Ngram datasets have been used for *culturonomics* [9] which is a study of the changes in word usage and cultural trends over time. In this work, we use *Google Books Ngrams* for the purpose of guessing document age and for reasoning about its temporal characteristics. To remove tokens generated as a result of OCR errors or those specific only to a particular document or

[1] http://www.gutenberg.org
[2] http://books.google.com
[3] http://www.archive.org

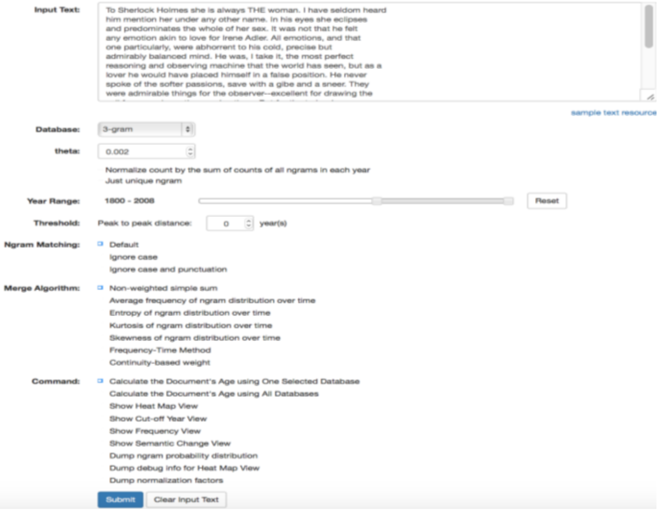
CIKM'17, November 6-10, 2017, Singapore.
© 2017 Association of Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11...$15.00.
DOI: https://doi.org/10.1145/3132847.3133166

[4] http://tinyurl.com/timestamping
[5] http://books.google.com/ngrams/datasets

author, we applied a threshold on the frequency of words in each decade equal to 300.

3. SYSTEM DESCRIPTION

Fig. 1 shows the main interface to the demo system which can be accessed through a web browser. The input text form at the top of the browser window is for entering the target text content. A user also has an option to use sample texts from the Internet Archive Text & Book Collection[6]. The time slider for setting the time range (denoted later as *T*) helps to limit the scope of age estimation, when the user wishes to analyze more closely the probability of text creation in a particular period. Users can select other parameters for estimating the text age such as number of grams, *n*, and θ (described further in Sec. 3.1 and 3.2), and then can choose one of the age detection methods from the "Merge Algorithm" section. Additional options (under the "Ngram matching" label in Fig. 1) let user decide whether word case and punctuations should be considered in ngram detection and matching steps. Finally, the "Command" section allows executing additional functionalities such as outputting the full list of ngrams together with their weights and occurrence counts in text.



Input Text: To Sherlock Holmes she is always THE woman. I have seldom heard him mention her under any other name. In his eyes she eclipses and predominates the whole of her sex. It was not that he felt any emotion akin to love for Irene Adler. All emotions, and that one particularly, were abhorrent to his cold, precise but admirably balanced mind. He was, I take it, the most perfect reasoning and observing machine that the world has seen, but as a lover he would have placed himself in a false position. He never spoke of the softer passions, save with a gibe and a sneer. They were admirable things for the observer--excellent for drawing the

Database: 3-gram

theta: 0.002

Normalize count by the sum of counts of all ngrams in each year
Just unique ngram

Year Range: 1800 - 2008

Threshold: Peak to peak distance: 0 year(s)

Ngram Matching:
- ● Default
- Ignore case
- Ignore case and punctuation

Merge Algorithm:
- ● Non-weighted simple sum
- Average frequency of ngram distribution over time
- Entropy of ngram distribution over time
- Kurtosis of ngram distribution over time
- Skewness of ngram distribution over time
- Frequency-Time Method
- Continuity-based weight

Command:
- ● Calculate the Document's Age using One Selected Database
- Calculate the Document's Age using All Databases
- Show Heat Map View
- Show Cut-off Year View
- Show Frequency View
- Show Semantic Change View
- Dump ngram probability distribution
- Dump debug info for Heat Map View
- Dump normalization factors

Figure 1. Snapshot of the main interface.

3.1 Age Estimation

After a user inputs the target text content in the provided text form[7] and sets the desired time frame of analysis, he or she can select the length of ngrams (option "Database" in Fig. 1) that will be used for age detection or he or she can choose an option to use all lengths of ngrams at once (*n={1,2,3,4,5}*). For a given *n*, all ngrams of length *n* will be extracted from the input text and matched to the underlying datasets. This is done by employing a sliding window(s) of length *n* over the input text. Each ngram found in the text is searched in the database so that the frequency plot of the ngram can be constructed over the selected time frame. The resulting plot is then normalized by first dividing the ngram count in each given year by the total sum of ngram counts contained in the dataset for this year. This step is carried to remove the effect of varying data sizes in different years[8]. Next, each normalized plot (one plot for each ngram) is converted to the probability distribution over the user-set time scope so that the sum of values from each year within the chosen time frame is equal to 1. The resulting distribution plots for all the ngrams extracted from the input text are then aggregated to compute the average plot. In particular, the final aggregate plot is the weighted average of the probability plots of all individual ngrams from the input text. There is also an option in the system to use only unique ngrams that occur in the target text. The score $S(y_i)$ in the merged plot for a year y_i is given by:

$$S(y_i) = \frac{\sum_j^M w_j P(t_j|y_i)}{\sum_j^M w_j} \quad (1)$$

M is the number of ngrams found in the input text, $P(t_j|y_j)$ is the estimated probability of an ngram t_j at year y_i as evaluated on the *Google Book Ngram datasets* according to the description shown above, and w_j are the weights decided by one of several possible choices to be defined bellow. The weights are based on the shapes of the ngram distribution plots. In other words, individual plots for given ngrams may count to varying extent when constructing the final distribution plot for the whole input text. The following options are provided:

Non-weighted simple sum: this option assumes equal weights (w_j=1) for aggregating plots of each ngram and is a default choice.

Average frequency of ngrams: here the weight of a given ngram is bound to the average frequency of the ngram in the dataset according to the intuition that the more common the ngram is, the more it should count for the age determination. Less frequent ngrams are susceptible to noise, hence, the frequent ngrams should have more reliable plots over time. We set the weight as follows: $w_j = logF(t_j)$ where $F(t_j)$ is the frequency of ngram t_j in the selected time period.

Entropy of ngram plots. High entropy ngrams are ones with probability distributions close to a uniform distribution. Such ngrams may not be discriminatory enough to estimate document age since they were commonly used by authors across many different decades. We then set high weights for the low entropy ngrams to give them preference in the age determination process: $w_j = H_{max} - \Sigma_N log(1/P(t_j|y_i))$. H_{max} is the maximum entropy found in the selected time period, while N is the number of years within the set time period.

Kurtosis of ngram plots. Similar to the entropy, the kurtosis measure prefers ngrams with distribution plots far from uniform (e.g., spiky plots that have high values for only few years). Entropy does not consider the number of peaks neither it can differentiate between ngrams whose plots have peaks close to each other (or rather far from each other). To reflect this intuition, we use kurtosis as another weighting method that favors ngrams characterized by distributions with one high peak. The weight is now given by (μ is a mean of the plot and σ is its standard deviation):

$$w(t_j) = \frac{\Sigma_N (P(t_j|y_i) - \mu)^4}{N\sigma^4} \quad (2)$$

Skewness of ngram plots. Skewness is another way to assign weight based on the ngram's plot shape. It gives high weight

[6] https://archive.org/details/texts

[7] The system also permits input through a batch mode where documents in .txt, .doc or similar format are taken and the age detection results (i.e., probability distribution of the creation date, and the final estimated date) are returned.

[8] In general, there is less data for years in the distant past than for ones in the more recent past.

values to ngrams which suddenly increased their frequency. Such ngrams can be useful for detecting the boundary of a document's age. Skewness is calculated as (ν denotes a mode):

$$S(t_j) = \frac{\mu - \nu}{\sigma} \tag{3}$$

Then the weight is given by:

$$w(t_j) = \begin{cases} 0 & if\ S(t_j) \le 0 \\ S(t_j) & if\ S(t_j) > 0 \end{cases} \tag{4}$$

Frequency-time method. The purpose of this method is to assign high weight to ngrams whose plots resemble step-like function. First, the years y_{min} and y_{max} when an ngram has its maximum frequency F_{max} and the minimum frequency F_{min}, respectively, are found in the selected time period *T*. Then the weight is:

$$w(t_j) = log\left(\frac{F_{max}}{F_{min}} * \frac{|T|}{|y_{max} - y_{min}|} + 1\right) \tag{5}$$

Continuity-based weight. It favors ngrams with a steady increase in their occurrence over time where the weight is computed as the sum of normalized differences between ngram probabilities in adjacent years.

The final merged plot for the input text is shown to the user in the graphical format as displayed in the example displayed in Fig. 2 (using *n*=3 in this case). We call it ***creation date probability distribution plot***. The year with the peak value of the plot is detected and proposed as the probable text's creation date. Fig. 3 shows five plots in a single view, each for a different value of *n*. As previously mentioned, computing and outputting results at once for all *n* values is also possible.

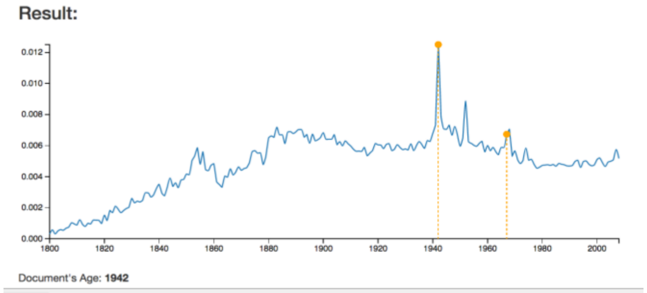


Figure 2. Creation date probability distribution plot and the detected year (1942) of the first part (809 words) of W. Churchill's speech "Address to Joint Session of US Congress, 1941"[9] based on 3-grams (non-weighted sum).

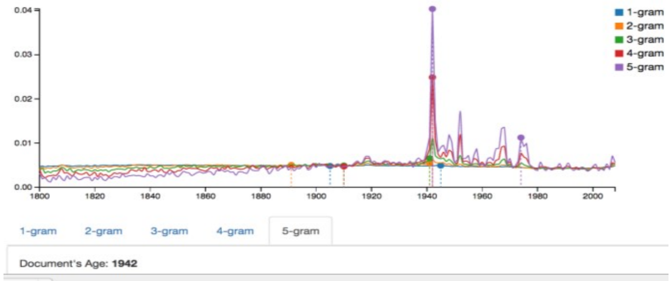


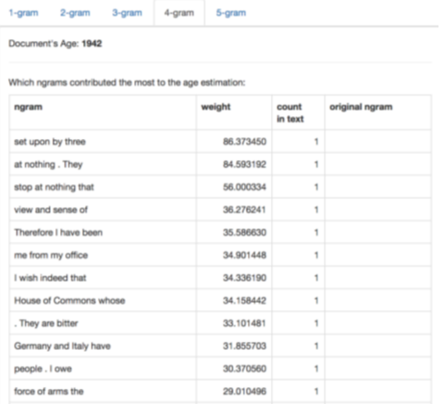
Figure 3. Creation date probability distribution plots for the sample text used in Fig. 2 based on ngrams for all *n* values (non-weighted sum).

3.2 Evidence Generation

Without a convincing proof automatic age estimation is not very useful, especially, for professional users who need sound evidence to trust machine generated answers. An important component of the system is then the evidence generation process. In order to provide reliable answers to the "why" question we propose various types of output. Users can analyze the returned evidences and "weight" their significance to reach the final conclusion. The following kinds of evidences are provided: (1) *Contributing ngrams*; (2) *Peak explanation*; (3) *Distribution of ngram boundaries over time*; and (4) *Document boundary*. These are described below.

3.2.1 *Contributing Ngrams*

The system can output the ranked list of ngrams together with the values of their weights to inform users *which ngrams contributed the most to the age estimation* when using Eq. 1. The actual weight score as well as the ngram occurrence are given to provide detailed information about the particular ngram's contribution to the estimated creation date. Example is shown in Fig. 4 for *n=4*.



Document's Age: **1942**

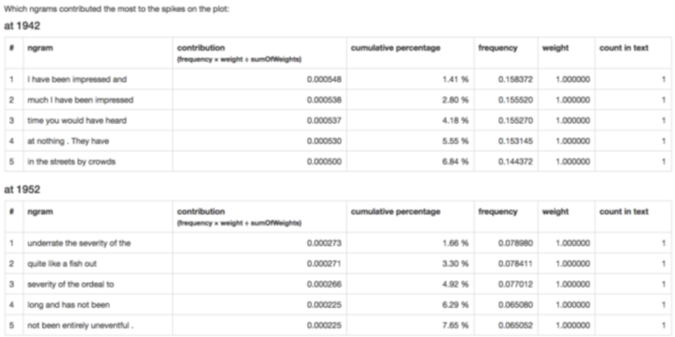
Which ngrams contributed the most to the age estimation:

ngram	weight	count in text	original ngram
set upon by three	86.373450	1	
at nothing . They	84.593192	1	
stop at nothing that	56.000334	1	
view and sense of	36.276241	1	
Therefore I have been	35.586630	1	
me from my office	34.901448	1	
I wish indeed that	34.336190	1	
House of Commons whose	34.158442	1	
. They are bitter	33.101481	1	
Germany and Italy have	31.855703	1	
people . I owe	30.370560	1	
force of arms the	29.010496	1	

Figure 4. Contributing top-scored 4-grams of the sample text used in Fig. 2.

3.2.2 *Peak Explanation*

Second, the top-5 peaks from the creation date probability distribution plot are found and the top-5 ngrams supporting each peak are shown (Fig. 5 shows such explanation for two peaks).



Which ngrams contributed the most to the spikes on the plot:

at 1942

#	ngram	contribution (frequency × weight ÷ sumOfWeights)	cumulative percentage	frequency	weight	count in text
1	I have been impressed and	0.000548	1.41 %	0.158372	1.000000	1
2	much I have been impressed	0.000538	2.80 %	0.155520	1.000000	1
3	time you would have heard	0.000537	4.18 %	0.155270	1.000000	1
4	at nothing . They have	0.000530	5.55 %	0.153145	1.000000	1
5	in the streets by crowds	0.000500	6.84 %	0.144372	1.000000	1

at 1952

#	ngram	contribution (frequency × weight ÷ sumOfWeights)	cumulative percentage	frequency	weight	count in text
1	underrate the severity of the	0.000273	1.66 %	0.078980	1.000000	1
2	quite like a fish out	0.000271	3.30 %	0.078411	1.000000	1
3	severity of the ordeal to	0.000266	4.92 %	0.077012	1.000000	1
4	long and has not been	0.000225	6.29 %	0.065080	1.000000	1
5	not been entirely uneventful .	0.000225	7.65 %	0.065052	1.000000	1

Figure 5. Explanation of the two highest peaks of the text sample used in Fig. 2.

For each peak, the top contributing ngrams are returned by analyzing the scores of their probability distributions in the years corresponding to the peak. The contribution is estimated by

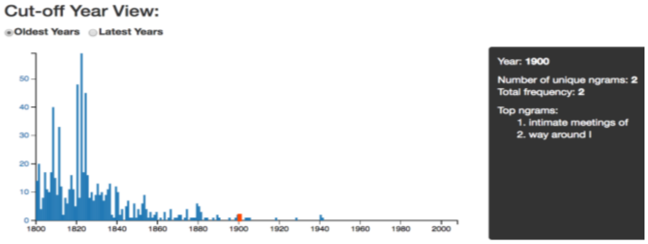
[9] www.nationalchurchillmuseum.org/churchill-address-to-congress.html

multiplying the frequency of the ngram in the input text with its weight and then by dividing it by the sum of all weights. The cumulative percentage of scores for the top ngrams is also shown. Since the peaks may sometimes be very close to each other (e.g., in adjacent years or separated by a few years only) the minimum allowed peak-to-peak distance can be set.

3.2.3 Distribution of Ngram Boundaries over Time

Next, a separate view[10] (see Fig. 6) displays the number of ngrams according to the oldest or latest years of time periods in which they were frequent. In particular, a user can choose two types of the plot in this view: "*by oldest year*" or "*by latest year*" depending on the type of information to be shown (i.e., the oldest and latest year of ngrams, respectively). The corresponding plot shows then on the vertical axis the number of unique ngrams which occur in the input text and which have their oldest or latest year falling into particular year on the horizontal axis. The *oldest year* of an ngram is defined as the year in which the ngram has been used for the first time with the frequency higher than the preset value of the parameter θ. On the other hand, the *latest year* is the one at which the ngram has been used for the last time with the frequency higher than θ. With a sufficiently low value of θ[11], the oldest and the latest years of an ngram can be considered as the boundaries of the time period when the ngram has been in a relatively common use. Naturally, sometimes ngram may have lower frequency than θ within that time period, however, for simplicity, we assume the continuity of ngram use within its left and right boundaries. This means that we reject the hypothetical situation when an ngram has been first commonly used in the past, then it become "forgotten" (i.e., unused) to later "re-emerge" again (subsequent frequent use).

When highlighting a given year (see Fig. 6), the system shows in a popup window the number of unique ngrams found in the target text that have the oldest (or the last) year falling into the selected year together with the data on the summed in-document count of these ngrams. The top k (k=5 by default) ngrams are also listed that have either the oldest (or the last) year equal to the investigated year. The ngrams are ordered by their in-document frequencies.



Cut-off Year View:

● Oldest Years ○ Latest Years

Year: 1900

Number of unique ngrams: 2
Total frequency: 2

Top ngrams:
1. intimate meetings of
2. way around I

Figure 6. Cut-off view with oldest years of the text used in Fig. 2 with the year 1900 being highlighted for detailed data.

3.2.4 Document Boundary

Finally, the boundaries of the text's probable age for different n values are shown on the creation date probability distribution plot (see dotted yellow lines in Fig. 2 and dotted lines for each different ngram plot in Fig. 3). These are selected as the minimum oldest and maximum latest years of all the ngrams found in the target text.

4. IMPLEMENTATION

To accommodate the large size of the *Google Books Ngram data*, we have used MapReduce framework, Apache Spark and PostgreSQL 9.3.9 with default indexing algorithm (B-tree). Scala 2.11.6 was utilized for data preprocessing and server-side programming together with the Web application framework: Play Framework 2.3.8. TypeScript 1.5 (JavaScript) was applied for client-side programming, while for UI we used the following libraries: D3.js 3.5.6, Bootstrap 3.3.2 and jQuery 2.1.3. Results are returned relatively fast, typically, within few seconds.

5. CONCLUSIONS

In this paper, we demonstrate an interactive online tool for facilitating document's age inference process as well as for supporting historical document understanding. The proposed system is the first of its kind and is available online to analyze arbitrary texts. In future, we plan to perform studies involving professionals and to consider different document genres. We will also include other features such as topic-level features as well as experiment with the named entity detection and disambiguation, which coupled with knowledge bases like Wikidata[12], can offer additional constraints for age inference.

ACKNOWLEDGMENTS

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013. It has been also funded by JSPS Grants-in-Aid for Scientific Research (#17H01828, #15K12158) and MIC/SCOPE (#171507010).

6. REFERENCES

[1] A.M. Ciobanu, A. Dinu, L.P. Dinu, and V. Niculae. Temporal Classification for Historical Romanian Texts. *LaTeCH2013*, pp. 102-106, 2013.

[2] F. De Jong, H. Rode, and D. Hiemstra. Temporal Language Models for the Disclosure of Historical Text. *AHC2005*, 161-168.

[3] A. Garcia-Fernandez, A.L. Ligozat, M. Dinarelli, D. Bernhard. When was it Written? Automatically Determining Publication Dates. *SPIRE2011*, pp. 221-236, 2011.

[4] N. Kanhabua and K. Nørvag. Improving Temporal Language Models for Determining Time of non-timestamped Documents. *ECDL2008*, pp. 358-370, 2008.

[5] N. Kanhabua and K. Nørvåg. Using Temporal Language Models for Document Dating, *MLKDD2009*, 738-741.

[6] D. Kotsakos, T. Lappas, D. Kotzias, D. Gunopulos, N. Kanhabua, and K. Nørvag. A burstiness-aware approach for document dating. *SIGIR2014*, p. 1003-1006, 2014.

[7] A. Kumar, M. Lease, and J. Baldridge. Supervised Language Modeling for Temporal Resolution of Texts. *CIKM2011*, pp. 2069-2072, 2011.

[8] W. Labov. Principles of Linguistic Change (Social Factors), Wiley-Blackwell, 2010.

[9] J.-B. Michel *et al.* Quantitative Analysis of Culture Using Millions of Digitized Books. Science, 331(6014), pp. 176-182, 2011.

[10] O. Popescu and C. Strapparava. SemEval 2015, Task 7: Diachronic Text Evaluation. *SemEval2015*.

[11] H. Salaberri, I. Salaberri, O. Arregi, and B. Zapirain. Ixagroupehudiac: A Multiple Approach System towards the Diachronic Evaluation of Texts. *SemEval2015*, pp. 840-845, 2015.

[10] Called "Cut-off year view" in the system.

[11] Currently, the value is set by the user. In the future we plan to offer automatic derivation of θ based on corpus-derived statistics.

[12] https://www.wikidata.org/