The RetroC challenge:

how to guess the publication year of a text?

Filip Graliński
Adam Mickiewicz University
Faculty of Mathematics and Computer Science
filipg@amu.edu.pl

Rafał Jaworski
Adam Mickiewicz University
Faculty of Mathematics and Computer Science
rjawor@amu.edu.pl

Łukasz Borchmann
Adam Mickiewicz University
Institute of Linguistics
borchmann@rainfox.org

Piotr Wierzchoń
Adam Mickiewicz University
Institute of Linguistics
wierzch@amu.edu.pl

ABSTRACT

This article describes research in automatic content-based temporal classification of texts. Experiments are carried out on a set of texts coming from Polish digital libraries, dating between the years 1814 and 2013. Following successful research in the field of temporal classification, this work aims at creating an automatic dating mechanism to be used in situations, where the publication date of the text is unknown. Automatic publication date assessment from the computer system can provide useful for researchers from various fields of humanities, such as history (incl. history of language), culture-historical archaeology, sociology or anthropology.

1. INTRODUCTION

Automatic content-based text classification is a known and often researched field of natural language processing. Specific classification problems include: sentiment analysis (where a text is assigned labels referring to the emotions it expresses), gender classification (where the problem consists in deciding whether a text is authored by a male or female) and many others. This article focuses on temporal classification, i.e. automatic prediction of the time the text was published.

In terms of machine learning algorithms, this problem can be viewed as regression. Publication time can be viewed as a continuous variable and with given training data any regression algorithm should be able to predict a specific point in time for any input text. However, this approach does not prove effective (as described in Section 2). Instead, the time is quantized into a series of intervals, often spanning one year. The problem is then defined as classification into these intervals. Some approaches (see for instance [3]) train different models for each time partition and for each new input text find the interval which is closest in terms of model distance measure.

This article describes a series of experiments in temporal classification on texts coming from Polish digital libraries. Importantly, this research assumes a methodology which allows for full transparency and reproducibility of results. In order to achieve that, the *Gonito.net* platform was used. The platform helped to organize the experiments in a form of a challenge for a group of researchers. Even though they were competing, the researchers had a full insight into all submitted solutions. This helped them gain inspiration and sparked fruitful discussions. As a result, we obtained a satisfactory solution which can be fully reproduced. Furthermore, we kept track of the thought process that led to this best solution.

The article itself is organized in the following way. Section 2 describes the background of this research. Training corpus used in the experiment is described in Section 3. Section 4 describes the experimental set-up and results achieved by different solutions to the problem. Conclusions are listed in Section 5.

2. RELATED WORK

This section is an overview of methods and mechanisms used in the field of temporal classification. Some of the most interesting solutions and experiments are described in detail.

2.1 AMBRA

AMBRA is a recently developed temporal classification system described in [4]. The system features some non-standard techniques and algorithms. Above all, the authors reformulated the problem of temporal classification of texts. Instead of predicting the exact point in time when the text was published, the output of the system is a time interval. Apart from that, the machine learning technique relies on training linear models reflecting time ranking of documents. These models are build on pairwise comparisons in the style of "what is the probability of document A being older than document B". A new input document is then predicted to be most probably older than e.g. document D_1 and most probably younger than document D_2, where both D_1 and D_2 come from the training data and their publication time is known.

DATeCH 2016 Poznań, Poland

Document features used by AMBRA to build the models are quite standard and include: length meta-features (number of sentences, types, tokens), stylistic (average word length, average sentence length, lexical density, lexical richness), grammatical (part-of-speech tag n-grams) and lexical (token n-grams). The authors engineered these features manually. Experiments revealed the following most informative features:

- Length of the text – this was by far the most informative feature. In the training data used, newer texts tended to be longer.
- POS n-gram: determiner + singular proper noun, characteristic for older texts
- POS n-gram: adjective + singular noun, characteristic for newer texts
- Informative words: highly topic-specific.

AMBRA achieved competitive results in the SemEval2015 shared task. However, it must be noted that the set of document features was prepared manually by the authors, which caused some of the features to be specific for the training data. For instance, the most informative feature was the length of the document, with newer texts being longer. This property can not be guaranteed for any training data.

2.2 Dalli and Wilks

[2] presented an approach to temporal classification that did not involve any manual feature engineering. Instead, the authors used an unsupervised machine learning approach to discover word repetitiveness patterns. Based on these patterns they inferred time series of word appearances in the texts and used them to build a model. Obtained model achieved high results in classifying news articles spanning the time period of 9 years.

This approach is probably especially well suited for news articles, as they tend to switch subject over time. By sheer analysis of words used in a newspaper article it may be concluded, which often reported event the article describes and hence to predict its publication time. However, this method could probably achieve satisfying results also for other types of training data.

2.3 de Jong, Kanhabua et al.

An early experiment on temporal classification was presented in [3]. The research was carried out on a data set of newspaper articles. Multiple experiments were carried out with different time partitions. Proposed classification algorithms relied on statistical methods and assumed two different approaches. The first approach consisted in comparing an unknown document to training documents in order to find the closest document in the training set. Prediction of the publication time of the unknown document was then based on the publication time of this closest document. Another approach assumed building separate models for each time partition. Interestingly, both classification algorithms were able to output not only the predicted publication time, but also a confidence score. Accuracy of predictions varied between 25%-55%, depending on required granularity.

Three years after this work, the article [9] was published. It described research that used the findings of [3] as baseline and managed to improve it. Novel techniques used in the improved approach included time span partitioning, so called word interpolation during feature extraction or temporal entropy (informative words were selected based on their entropy).

2.4 HTRC

Both [12] and [8] describe the experiment on temporal classification of digital library data. The corpus contained 250 000 volumes from the HathiTrust's digital library, dating from the year 1600 to 2000. The authors designed a temporal classification algorithm based on logistic regression, support vector machines and decision trees. Statistical features were engineered manually and included:

- first date mentioned in text
- text similarity metrics combining n-grams
- OCR error counts

Extracted features of each document are used to build document vectors. In order to predict the publication time of an unknown document, its vector is compared to so called chronon vectors, i.e. vectors representing specific points in time. This approach achieved an F-score of 0.86 in the experiments.

2.5 Research in Romanian historical texts

The article [1] describes a temporal classification algorithm for Romanian historical texts. The algorithm is based on linear support vector machines and random forests. Statistical features were engineered manually and are largely based on dictionaries. However, it must be noted that this system works on a granularity of 100 years, i.e. is only able to predict the century when the input document was published.

2.6 Kumar et al.

A non-standard approach to temporal classification is presented in [10]. The article describes research on predicting not the publication time of the text, but the time of the story the text tells. The algorithm used to solve this problem builds a statistical model of histograms of probabilities of a text belonging to a specific period. Classification itself is based on Kullback-Leibler Divergence between test document model and models for periods. Experiments involved a corpus of 678 Gutenberg short stories, enhanced with helping data from Wikipedia biographies.

3. TRAINING DATA

Training data used in the experiments presented in this article comes from the RetroC corpus [7]. It is a corpus of Polish texts from digital libraries dating between 1814 and 2013, enriched with old textual material from other online sources.

As the size of a data set is essential for performing statistical analysis, creators of RetroC intended to collect a vast amount of texts. The corpus contains fragments of texts of exactly 500 words. They are split into 3 subsets: training set (40 000 fragments), development set (9 910 fragments) and test set (10 000 fragments). The development and test sets are balanced with respect to the principle: take exactly 50 publications per year. However, training set did not undergo any balancing procedures.

Since most of the text in the RetroC corpus is the output of an OCR algorithm, the text is not free from noise. Only minimal preprocessing was applied, aimed at, among others, joining words separated by hyphens, removing excessive newline characters and UTF sanitation. No time expressions (especially explicit dates) were removed from the texts, as they are to be used as highly informative features for learning algorithms.

4. EXPERIMENTS AND RESULTS

4.1 Experimental setup

All experiments on temporal classification described in this article were carried out with the help of the *Gonito.net* platform [6]. It is open source software allowing for easy management, reuse and reproducibility of research.

The task of temporal classification of the texts described in Section 3 was organized as a challenge for a group of researchers. The challenge was formulated as follows: "Guess the publication year of a Polish text.". Each researcher (referred to as competitor) was given the entire training and development sets, while the test set was hidden. Competitors then developed their classification mechanisms and tested them on the development set. For the sake of evaluation of performance of each solution, the root-mean-square error metric (RMSE) was used.

Evaluation of each solution was performed in 2 stages. First, RMSE was computed based on the performance of the mechanism on the development set. This evaluation could also be performed by the competitor, as he possessed all the needed data, i.e. the whole development set. Then, each submitted solution was tested automatically by *Gonito.net* on the test data. The RMSE on the test corpus was the only score taken into account while preparing the ranking of solutions.

At any time the competitors have access to the current leader board, presenting best solutions submitted so far. This setup encourages competitiveness and increases the motivation. However, mere scores achieved by other researchers are not the only information about the solutions published by *Gonito.net*. The platform is based strictly on Git. Each challenge is defined in its own repository, which is branched by the competitors. Each solution is identified with a commit in a competitor's branch of the original challenge repo. Thanks to that architecture, the source code of each solution can be easily tracked and retrieved by *Gonito.net*. Competitors are therefore able to inspect other solutions and improve them, if they see such possibility. Such experimental setup allows for research transparency and reproducibility, leading to achieving optimal final results.

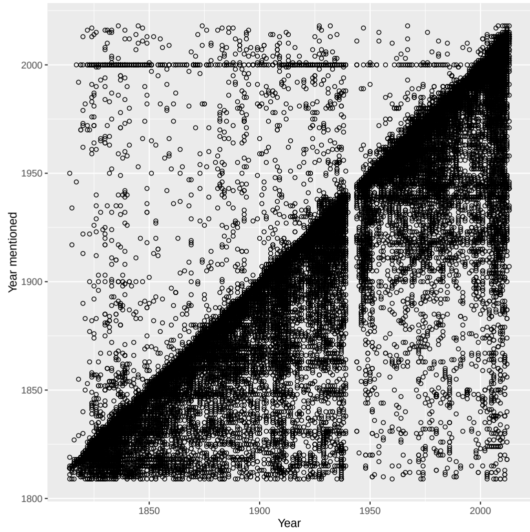
The following subsections describe solutions for the problem of temporal classification of Polish texts developed by our group of researchers with the help of the *Gonito.net* platform. For some of the solutions we publish a so called gonito.net url, pointing to the specific commit with the source code of the solution. The url has the following shape:

`http://gonito.net/q/<commit-id>`

Such urls can be used to obtain the repository of the solution, reproduce its results and possibly improve them.

4.2 Null model baseline

Figure 1: Year references



As the first naive baseline we assumed a mechanism that always returns the median year 1913. This solution yields RMSE of **57.9** years on the development set. The null model is published on *Gonito.net* as commit:
`http://gonito.net/q/d80206`.

Naturally, we expect all subsequent solutions to achieve substantially better scores. In the below subsections we present all notable solutions, i.e. those that managed to achieve results considerably above the null model, but also those which, surprisingly, fell below expectations.

4.3 Year references

A simple tool was created to extract year references from RetroC texts – it was based on a regular expression for the time span in question ± 5 (1809–2018) plus a blacklist of words which cannot follow a year reference (e.g. monetary units). Of the items in the training set 52.1% do not contain any year reference, 17.9% contain exactly one year reference, and 30.0% contain more than one. The distribution of year references vs. publication years is shown in Figure 1.

Based on observation of the figure, the following simple baseline for automatic dating may be proposed:

- if no year reference is found, return 1913 (median year);
- if only references to year 2000 are found, return 2000;
- otherwise, return the latest year reference other than 2000.

This method yields RMSE = **45.9** years on the dev set (vs RMSE = 57.9 for the null model). Commit id for this solution is: `http://gonito.net/q/fa3215`.

4.4 OCR noise

It might be assumed that the older the text, the lower the quality of the OCR output (and the higher the probability of finding words written with obsolete orthographic conventions, archaic words, etc.). In order to check this assumption, we counted the percentage of words recognized by a modern spell-checker.

Unfortunately, the linear regression model with respect to the logit of these percentages yields the same RMSE as the null model.

4.5 Polish orthography

Because of the country's turbulent history, the study of Polish orthography covers a whole range of issues related to frequent reforms, geographical division, fragmentation of the academic community, the preferences of publishers and users of the language, and so on. Describing the changes in and practical application of the rules of Polish orthography is an extremely complex task. Here we will attempt to summarize this topic in brief.

At the start of the Modern Polish era, matters of orthography were taken up by Onufry Kopczyński. He reintroduced the letters *á* and *é*, although their usage was very difficult. In 1816, Alojzy Feliński rejected the use of the character *á*, while retaining *é* and *ó*. In place of the formerly used *i* and *y* he introduced *j* (*jajko* instead of *iayko*).

In 1890 the Academy of Learning set up an Orthographic Commission, which in 1891 published a set of rules prescribing:

- the forms *módz*, *biedz* (rather than *móc*, *biec*);
- *ge* rather than *gie*, for example *geografia* rather than *gieografia*;
- *Francya*, *Anglya* (not *Francja*, *Anglia*), *Maryja*, *Julia* (not *Marja*, *Julja*), *kolacya*, *dyalekt*, *dyagnoza*;
- the endings *-im*/*-ym* as alternatives to *-em*, and *-imi*/*-ymi* as alternatives to *-emi*.

These reforms met with serious objections. Consequently, in 1906, the Linguistic Commission of the Academy of Learning approved the following principles:

- *-ja* in place of *-ia*, *-ya*, but only in final syllables (e.g. *biologja*, *diatermja*);
- *gie* in place of *ge*, as in *gieografja*, *gienerał*;
- the endings *-ym*, *-im* and *-ymi*, *-imi* (alternatives to *-emi*);
- the infinitive forms *biec*, *móc*.

Further innovations followed in 1918. The changes announced in the Main Principles of Spelling included the following:

- in non-initial syllables, in foreign words, the letter *j* to be used after a consonant;
- the endings *-em*, *-emi*, *-ym*, *-ymi* to remain distinct according to the ending of the nominative;
- foreign words to be written with *ke*, *ge*, but *kie*, *gie* to be used in words felt to be native;
- *-c* to be used in infinitives of the type *biec*, *móc*.

These rules were not universally adopted, and moreover they sparked controversy. Hence, in 1932 a new set of rules appeared, although these related mainly to questions of spacing. For example, it was decided that expressions consisting of a preposition and a noun would be written as a single word if they had an adverbial meaning, as *pomału*, *zamłodu*, *zbliska*, *nakoniec*.

	1814–1918	1918–1936	1936–2013
-dz ending	1.0	0.5	0.0
cya	1.0	0.5	0.0
rya	1.0	0.5	0.0
rja	0.5	1.0	0.0
-emi ending	1.0	1.0	0.0
-ymi ending	0.0	0.0	1.0

Table 1: Scores for specific variants

The greatest reform of the orthographic system was enacted in 1936:

- words like *Maria* to be written with *i*, except after *c*, *s*, *z* (e.g. *Francja*, *pasja*, *diecezja*);
- inflectional endings of adjectives to be standardized as *-ym*, *-ymi*;
- *ke* in foreign words to be written *kie*, but foreign *ge* not as *gie*;
- the negating particle *nie* should not be written as one word together with participles having a verbal meaning;
- changes were made to the spellings of certain specific words, such as *brózda* → *bruzda*, *chróst* → *chrust*.

Taking into account this knowledge, we decided to partition the whole time span into the three periods 1814–1918, 1918–1936 and 1936–2013, score the periods according to Table 1, and take the median year in the winning period. This method results in recall 58.8%, precision 83.1% (for recognition of one of the three periods), and RMSE = **46.6** (dev set).

4.6 External sources

As in the previously mentioned work [10] we experimented with using external resources to help in the temporal classification problem. We extracted the birth years of 210,297 people from Polish Wikipedia (cf. [5]). Unfortunately, it turned out that only 12.7% of the training set contained references to any of these people, and the level of noise was quite high – 19.3% of matches appeared in fragment dated earlier than the corresponding birth year (even though both the first name and the surname were required for a match). It is not surprising, then, that the predictor based only on birth years is no better than the null model (RMSE = 57.4, dev set).

4.7 Vowpal Wabbit

Temporal classifiers can be trained with supervised learning methods (supply a collection of texts annotated with dates, and apply a machine learning classification or regression algorithm). So far, classification approaches have been used (create a separate model for each year, and for a given text assign the year for which the classifier yielded the highest probability or score); see e.g. [5]. Here, we will consider the regression approach: one model with the output (year) taking continuous values (given with any precision, not necessarily rounded, i.e. we accept values such as 1951.440777).

We used the Vowpal Wabbit open-source learning system [11]. As features we simply used lower-cased tokens and/or character pentagrams (as suggested in [5]); the predicted value is $y - 1913.0$ for year y, and the number of training passes was 40. In order to account for the difference in distribution between the training set and the development/test set, we applied inverse weighting of training examples (with the log function).

Using pentagrams yielded a better result (RMSE = 22.0, dev set) than using tokens (RMSE = 22.8, dev set). This is not surprising, as character-level n-grams are more robust to OCR noise, and they are a simple yet effective substitute for lemmatisation in the case of inflected languages such as Polish. Using both tokens and n-grams did not improve the quality of the system (RMSE = 21.9, dev set).

A natural idea is to add the parameters discussed in previous sections as features. Interestingly, neither adding the year extracted as described in Section 4.3 (RMSE=21.9, dev set) nor knowledge about changes in orthography (Section 4.5, RMSE=21.8, dev set) improves the quality of the system significantly. If both of these types of features are added, we obtain RMSE=21.7 on the development set.

4.8 Wikipedia years

In another experiment, we decided to exploit Wikipedia in order to extract temporal information from the contents of the articles. We assumed the following approach:

1. Prepare a "Wikipedia tagging mechanism" which is able to efficiently find all occurrences of Wikipedia article titles in a text.
2. For each input document, apply the tagging mechanism and extract a list of Wikipedia articles, whose titles were mentioned in the document.
3. For each of those articles, extract all year references mentioned in the article.
4. Take the median of all years extracted in such manner and return it as predicted publication time of the document.

Unfortunately, this approach does not yield good results in its current version. The predictions tend to be considerably too high, as Wikipedia articles tend to contain many contemporary year references, among others in the "In pop-culture" section. More experiments are planned, implementing different strategies of extracting years from the Wiki articles.

4.9 Vowpal Wabbit with neural networks

A significant breakthrough and currently the best result was achieved by running the Vowpal Wabbit regression tool with the option of neural networks unsupervised learning. The solution was based on the following set of document features:

- all 1-, 2-, 3- and 4-grams of letters,
- tokens of at least 5 letters, truncated to 7 letters

For each training document, these features are computed along with the weight score. The weight is computed for the whole document on the base of token frequencies within years. If a document contains a large number of words which appeared frequently in the year this document was published, such document gets a low weight. This is to avoid vocabulary bias.

A neural network operating on 6 neurons is used as an additional helping mechanism. In order to determine the optimal number of neurons, a hyper-optimization technique from the Vowpal Wabbit toolkit was used. Another important Vowpal Wabbit attribute – number of bits in the feature table – was originally set to 29 which made the models require approximately 16GB of RAM. However, it was later checked that reducing this number to 25 brought the size of the model down to 1GB, while keeping the same results.

The overall scores for this technique were 17.2 on the development set and 24.8 on the test set. [1] This solution is published on *Gonito.net* as commit:
`http://gonito.net/q/9dcf6a`

method	dev	test-A
null model	57.9	57.7
year references	45.9	46.4
orthography	46.6	50.9
5-grams	22.0	33.5
5-grams + year ref. + ortho.	21.7	33.1
5-grams + year ref. + ortho.	21.7	33.1
vw + nn	**17.2**	**24.8**

Table 2: Summary of results

4.10 Results summary

A summary of the results obtained on both the dev set and the test set is presented in Table 2.

Figure 2 presents the current leaderboard, showing the results on the test set (test-A).

Acknowledgments

Work supported by the **Polish Ministry of Science and Higher Education** under the **National Programme for Development of the Humanities**, grant 0286/ NPRH4/ H1a/ 83/ 2015: "50 000 słów. Indeks tematyczno- chronologizacyjny 1918-1939".

5. CONCLUSIONS

We presented a series of experiments on temporal classification of text documents. Though varying in terms of approach, all experiments were run on the same test data obtained from Polish digital libraries. The *Gonito.net* platform significantly facilitated the process of publishing the test data for researchers, developing and evaluating solutions, as well as provided means for research reproducibility. Many of the submitted solutions were largely inspired or physically based on other solutions. Furthermore, *Gonito.net* provided additional motivation to the researchers by formulating the temporal classification problem as a challenge.

In terms of the results achieved, we conclude that the best solution, achieving RMSE = 24.8 on the test set may

[1] Interestingly, a very similar result: test-A = 24.9 was obtained by using a pure neural network approach. This experiment, however, was not fully transparent.

Figure 2: Gonito leaderboard

Leaderboard

#	submitter	when	description	test-A/RMSE	×
1	p/tlen	2015-12-13 14:31	VW -nn 6 on up to 4-grams and [5-7] tokens	24.8	18
2	Marcin Junczys-Dowmunt	2015-12-12 22:02	The same as last, best epoch	24.9	11
3	[anonymised]	2016-04-30 18:20	Solution for 4grams and 40 class	40.3	8
4	[anonymised]	2015-12-16 20:13	+ script	50.3	3
5	[anonymised]	2015-12-16 23:38	first submission	75.0	1

be viewed as a success. This best solution relied solely on the training data, with no external data sources. So did the mentioned pure neural network solution, coming from a researcher from outside of our team, who achieved a very similar result of 24.9. It might be the case that no significantly better results can be achieved by analyzing only the training data. Therefore, future work will concentrate on exploiting external knowledge sources to furtherly improve the temporal classification results.

6. REFERENCES

[1] A. M. Ciobanu, A. Dinu, L. P. Dinu, V. Niculae, and O.-M. Sulea. Temporal classification for historical romanian texts. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 102–106, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[2] A. Dalli and Y. Wilks. Automatic dating of documents and temporal text classification. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 17–22. Association for Computational Linguistics, 2006.

[3] F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. In *Humanities, computers and cultural heritage: Proceedings of the XVIth International Conference of the Association for History and Computing (AHC 2005)*, pages 161–168, Amsterdam, The Netherlands, September 2005. Royal Netherlands Academy of Arts and Sciences. Imported from EWI/DB PMS [db-utwente:inpr:0000003683].

[4] M. E. Z. de Marco, A. M. Ciobanu, V. Niculae, and L. P. Dinu. Ambra: A ranking approach to temporal text classification. In *Proceedings of the 9th Workshop on Semantic Evaluation (SemEval)*. ACL, 2015.

[5] A. Garcia-Fernandez, A.-L. Ligozat, M. Dinarelli, and D. Bernhard. When was it written? automatically determining publication dates. In *String Processing and Information Retrieval*, pages 221–236. Springer, 2011.

[6] F. Graliński, R. Jaworski, Łukasz Borchmann, and P. Wierzchoń. Gonito.net – open platform for research competition, cooperation and reproducibility. In A. Branco, N. Calzolari, and C. Khalid, editors, *Proceedings of the Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pages 13–20, 2016.

[7] F. Graliński and P. Wierzchoń. RetroC – A Corpus for Evaluating Temporal Classifiers. In Z. Vetulani and J. Mariani, editors, *Proceedings of 7th Language & Technology Conference*, pages 245–249, Poznań, 2015. Fundacja Uniwersytetu im. Adama Mickiewicza.

[8] S. Guo, T. Edelblute, B. Dai, M. Chen, and X. Liu. Toward enhanced metadata quality of large-scale digital libraries: Estimating volume time range. In *Proceedings of the iConference 2015. March 2015, Newport Beach, California.*, 2015.

[9] N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In B. Christensen-Dalsgaard, D. Castelli, B. A. Jurik, and J. Lippincott, editors, *ECDL*, volume 5173 of *Lecture Notes in Computer Science*, pages 358–370. Springer, 2008.

[10] A. Kumar, M. Lease, and J. Baldridge. Supervised language modeling for temporal resolution of texts. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2069–2072, New York, NY, USA, 2011. ACM.

[11] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. In *Advances in neural information processing systems*, pages 905–912, 2009.

[12] B. Plale and M. Chen. Predicting publication date: a text analysis exercise over 250,000 volumes in the htrc secure hathitrust analytics research commons. RDA Digital Humanities Workshop, May 2015, 2015.