Simple Neologism Based Domain Independent Models to Predict Year of Authorship

Vivek Kulkarni
Department of Computer Science
University of California, Santa Barbara
vvkulkarni@cs.ucsb.edu

Yingtao Tian
Department of Computer Science
Stony Brook University
yittian@cs.stonybrook.edu

Parth Dandiwala
Department of Computer Science
Stony Brook University
pdandiwala@cs.stonybrook.edu

Steven Skiena
Department of Computer Science
Stony Brook University
skiena@cs.stonybrook.edu

Abstract

We present domain independent models to date documents based only on neologism usage patterns. Our models capture patterns of neologism usage over time to date texts, provide insights into temporal locality of word usage over a span of 150 years, and generalize to various domains like News, Fiction, and Non-Fiction with competitive performance. Quite intriguingly, we show that by modeling only the distribution of usage counts over neologisms (the model being *agnostic* of the particular words themselves), we achieve competitive performance using several orders of magnitude fewer features (only 200 input features) compared to state of the art models some of which use 200K features.

1 Introduction

Determining when a document is written is an important task in historical linguistics and temporal information retrieval. For instance, several works attempt to date historical biblical texts like the *The Book of Isiah* (Rooker, 1996; Ehrensvärd, 1997; Hurvitz, 2000; Young and Rezetko, 2016) or ancient texts like *Beowoulf* (Chase, 1997). Likewise, in the field of information retrieval, establishing the dates of documents is an important pre-requisite to returning temporally relevant documents and provides important information for a large number of search tasks (Ostroumova Prokhorenkova et al., 2016; Efron, 2013).

Most efforts to automatically date texts adopt a learning based approach and rely on several linguistic features that are time-relevant (Garcia-Fernandez et al., 2011; Jatowt and Tanaka, 2012; Zampieri et al., 2015; Ostroumova Prokhorenkova et al., 2016). Such time-relevant features include neologisms/archaisms, political events, spelling variations, and the presence of named entities as well as external knowledge bases. In addition to the large input feature dimensionality (of the order of the input vocabulary), these approaches all focus on particular domains (for example, primarily News articles). While such feature rich models which are arguably domain specific perform very well, these features are rarely generalizable across domains (for example. models that use political events can perform very well when evaluated in the News domain, but do not necessarily generalize to Fiction). Developing models that generalize across domains without further fine tuning generally entails modeling linguistic cues that are stable across domains.[1]

Consequently, in this work, we propose the first set of models that effectively generalize across domains by leveraging insights into the temporal usage of language. Our models rely on a key insight which we term the **temporal locality of neologisms**: *Documents written at time t tend to use neologisms invented shortly before t.* By effectively modeling this usage of neologisms (as a feature class cumulatively) we propose models that not only reduce our input feature dimensionality by at-least an order of magnitude, but also effectively generalize across a variety of domains (Fiction, Non-Fiction, and News)

[1] It is worth noting that ultimately most domain adaptation techniques attempt to find such a general feature set by either attempting to map source domain features to the target domain or learning a more abstract feature set.

Proceedings of the 27th International Conference on Computational Linguistics, pages 202–212
Santa Fe, New Mexico, USA, August 20-26, 2018.

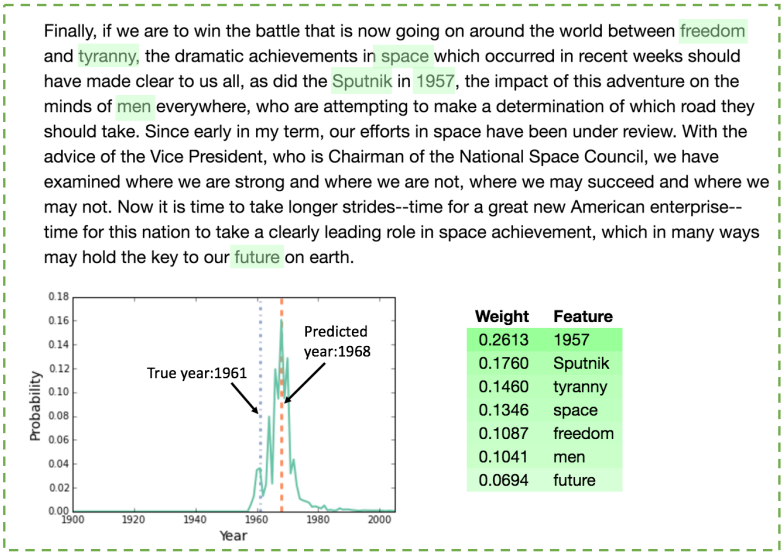


Figure 1: Sample output predictions of a Naïve Bayes model (NB) on a portion of a speech given by President John F.Kennedy in 1961. Note that this model outputs a probability distribution over years with a MAP estimate of year 1968. Note also that the model was trained *only using* Google Book Ngrams and not on the domain it is evaluated on. Finally, observe that words like `1957, Sputnik, tyranny and space` were most influential in this prediction thus providing insight into linguistic patterns the model has captured only from Google Book Ngrams. Moreover, the model is generic and can be applied to multiple domains like Fiction, News or Non Fiction. This motivates our hypothesis that neologisms can be effectively used to date documents across multiple domains.

without further tuning and perform competitively with more complex models that capture fine-grained linguistic cues. Intriguingly, we demonstrate that neologism-based models that use only ~ 200 features achieve a performance within 5 units of mean absolute error (21.58 on NonFiction) over the best Naïve Bayes model (18.25 on NonFiction) which uses more than 200K features.

In a nutshell, we make the following contributions:

- We propose domain independent models for the task of dating documents. We emphasize that our goal is not to necessarily outperform the state of the art domain specific models, but to demonstrate the effectiveness of simple models drawing on linguistic insights that generalize across domains.

- We leverage cumulative usage patterns of neologisms over time to propose the first set of simple generalizable models for this task while revealing insights into the cumulative usage patterns of neologisms over time.

- We empirically evaluate our models against several competing methods including neural models like LSTMs on three different domains (News, Fiction and Non Fiction).

2 Datasets

Here, we describe data-sets which we use for learning and evaluating our models.

Training Dataset We train all of our models using only the Google Book Ngrams dataset spanning the time $1850 - 2005$. Since the Google Book Ngrams spans multiple domains, we can capture domain agnostic linguistic cues to learn generalizable models for our task. Since our neologism models need only the frequency of occurrences, the Google Book Ngrams is an ideal dataset providing not only a large sample size for robust parameter estimation but also inherently spanning multiple domains.

Evaluation Datasets To evaluate our models, we consider the following datasets[2]:

- **NYTimes**: We consider a random sample of 10000 leading paragraphs of NEW YORK TIMES articles from the range $1850 - 2005$ constructed by scraping the New York Times website. Note that this dataset is primarily from the NEWS domain.
- **Corpus of Historical American English (COHA)**: We consider a random sample of 10000 articles from each genre, namely FICTION, NONFICTION, and NEWS from the COHA corpus (Davies, 2002). The COHA corpus is an ideal dataset to evaluate our models since it spans a wide time range, with multiple domains where the dates have been validated by human experts and is easily available for research purposes[3].

We emphasize that all of our models only use the Google Book Ngrams data to learn parameters. The models are then evaluated on the evaluation datasets which span multiple domains without any further fine-tuning.

3 Baselines

Before we describe our proposed models, we introduce two baseline methods to evaluate against on our task.

BOOKPROP We estimate the probability of a document written in a given year y, by computing the fraction of books written in year y over all books written in the time period under consideration. Formally, we estimate the following probability:

$$\mathsf{P}(Y = y) = \frac{\#(\text{books}, y)}{\sum_y \#(\text{books}, y)}$$

We estimate the number of books in English written in year y, as the number of distinct books the word `the` was mentioned in a given year y as per Google Book Ngrams (Michel and others, 2011) data. As expected, the distribution is skewed towards the right with more books written in the 20^{th} century than in the 18^{th} century (see Figure 2). Given a document to date, a random sample drawn from this distribution is then taken as the predicted estimate of the date of the document. A limitation of BOOKPROP is that it does not model language.

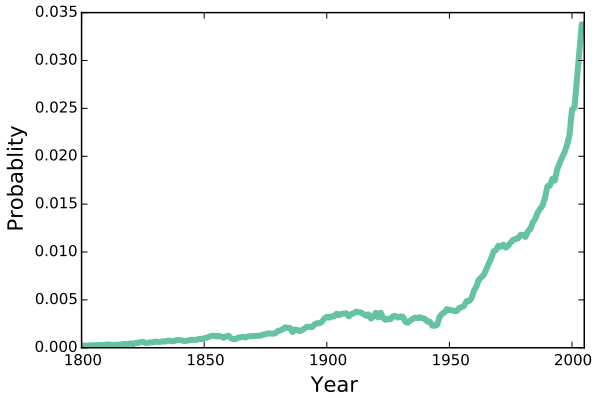


Figure 2: Estimate of the probability of English books being from a given year using the Google Book Ngrams data. As expected, the distribution is skewed towards the right with more books written in the 20^{th} century than in the 18^{th} century.

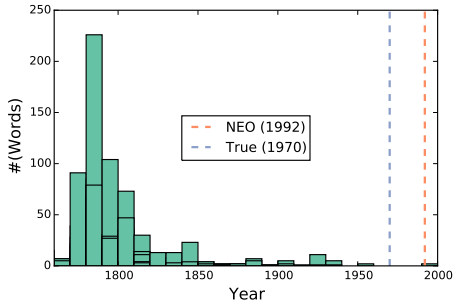
[2]While there has been work on dating documents (Zampieri et al., 2016; Kumar et al., 2012; Graliński et al., 2017), there are no standard publicly available evaluation datasets in the community for this task especially spanning multiple domains in English.

[3]We note that another option to construct such a dataset spanning multiple domains would be to use books from the HathiTrust. However, accessing a large clean dataset required institutional access unavailable to us at this point in time.

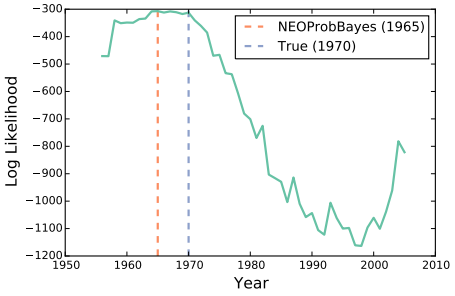
WORD	Estimated Year	Actual First Usage
HIV	1987	1986
Hitler	1933	1934
LSD	1955	1950
Obama	2007	2006
SARS	2003	2003
⋮	⋮	⋮

WORD	Estimated Year	Actual First Usage
Sputnik	1958	1957
electron	1905	1891
radio	1904	1907
television	1931	1907
transistor	1950	1948
walkman	1993	1979

Table 1: Example cases of estimated year of popular usage (MR) and actual year of first use (FU) obtained from `http://www.etymonline.com/` for different words from Google Book Ngrams data. Note that in the majority of these words, our estimated year is close to the year of first usage and shows a small lag from the year of first use as expected.



(a) NEO



(b) NEOProbBayes

Figure 3: Figure illustrating NEO and NEOPROBBAYES on the same document which was written in 1970. NEO is easily misled by outlier words and predicts the date to be 1992 ignoring other evidence like counts of other words. NEOPROBBAYES, in contrast incorporates all observed evidence to estimate more accurately that the document was written in 1965.

NEO A more sophisticated approach to assigning dates to documents is based on the following observation: If we observe a word which first came into popular usage in a year y, then the document is very likely written after year y. A simple model based on this hypothesis is to output the year of the most recent word found in the document. For example, in Figure 1, NEO estimates the date of a document to be 1958, since it is clear that `Sputnik`, the most recent word used in the document, sprung into popular use in 1958.

We estimate the year in which a word came into popular usage MR(w) from Google Book Ngrams as follows: (a) Compute the cumulative usage of a word w through every year in the Google Book Ngrams. (b) Compute the first year in which the cumulative usage of the word w exceeds a small fraction α (empirically set to $1/250.0$) of the total cumulative usage. As an example, our method estimates MR(`Obama`) $= 2007$ while the year of actual first usage is 2006. We validate our approach on a small set of manually curated words which we show in Table 1.

Observe that for the majority of these words, our estimated year is close to the year of first usage and shows a small lag, which is expected since we seek to estimate the year in which the word came into popular usage and not its year of first usage.

While NEO serves as a strong baseline, there are two limitations of this method: (a) The document could be written long after the time period corresponding to the most recent word observed in the document. (b) It ignores evidence of other words seen in the document and bases its decision on the occurrence of a single word.

Figure 3a illustrates these drawbacks. First, note that only one word `vinaya` with an MR(w) in the 1990's was observed in this document. NEO is easily misled by this single erroneous estimation of MR(`vinaya`) and estimates the date of this document to be 1992. It ignores other evidence that suggests that the document was written after 1940, but is unlikely to be written in the 1990's since words

with $\mathrm{MR}(w)$ a few decades before 1990's are not observed at all.

4 Proposed Neologism Based Model

We now describe a probabilistic model that effectively uses new words incorporated into popular usage to estimate when the document was written. In particular, our model computes the likelihood of observing a set of words that came into popular usage after year x given the document was written in year y to estimate when the document was written. Our method has two key steps:

1. **Ensemble Model Construction**: We construct an ensemble of probabilistic models where model M_i outputs $\mathsf{P}(y|X_i)$ and X_i is a discrete random variable counting the words observed in a document which came into popular usage after year i.
2. **Combining Ensemble Predictions**: Each model M_i outputs $\mathsf{P}(y|X_i)$, so we investigate multiple methods to combine predictions from individual models.

Ensemble Model Construction Let $\mathsf{F}(o, n)$ be the probability of observing a word that came into popular usage after year o in year n, where $n > o$. For every year pair (o, n), we estimate $\mathsf{F}(o, n)$ from the Google Books Ngrams Corpus by computing the fraction of words with $\mathrm{MR}(w) > o$ in the Google Book Ngrams of year n.

Given a text T of length N, let $N'(i)$ denote a realization of X_i in T. Each model M_i models the probability of T written in year y based on X_i as follows:

$$\mathsf{P}(y|X_i) \propto \begin{cases} \mathsf{P}(X_i|y)\mathsf{P}(y), & \text{if } i < y \\ 0, & \text{otherwise} \end{cases}$$

$\mathsf{P}(X_i|y)$ follows a binomial distribution with success probability $\mathsf{F}(i, y)$ which can be computed knowing the length of the document N and $N'(i)$ a realization of X_i. We assume the prior $\mathsf{P}(y)$ to be uniform.

Combining Ensemble Predictions Each model M_i computes a probability distribution over years, namely $\mathsf{P}(y|X_i)$. We now describe three methods to combine these individual model predictions to output a final prediction:

1. **NEOProbMean**: We output the mean of individual MAP predictions as the predicted year of authorship.
2. **NEOProbMedian**: We output the median of individual MAP predictions as the predicted year of authorship.
3. **NEOProbBayes**: We use a Bayesian scheme to incorporate all the observed evidence as follows: Let $\boldsymbol{X} = \{X_i \text{ for each year } i\}$. Specifically we compute the following:

$$\begin{aligned} \mathsf{P}(y|\boldsymbol{X}) &\propto \mathsf{P}(\boldsymbol{X}|y)\mathsf{P}(y) \\ &= \left(\prod_i \mathsf{P}(X_i|y)\right)\mathsf{P}(y) \end{aligned}$$

   where we make the *Naïve Bayes assumption* that each X_i is independent of any other X_j, when conditioned on the year y. We output the MAP estimate of $\mathsf{P}(y|\boldsymbol{X})$ as the final prediction.

Figure 3b shows this approach for a document and also contrasts it with the baseline NEO. Observe how NEOPROBBAYES enables a more accurate prediction by incorporating observed evidence ignored by NEO.

As we will show empirically, performance of each of the above methods is a function of the length of the document it is evaluated on since the accuracy of the individual probability estimates depends on the length. [4] While NEOPROBBAYES is better for large documents, it is quite sensitive to errors in estimates for small documents. Therefore, NEOPROBMEDIAN and NEOPROBMEAN which are less sensitive in the presence of outliers are better than NEOPROBBAYES for short documents (100 tokens) but are outperformed by NEOPROBBAYES for larger documents (2000 tokens) when individual model estimates are much more accurate.

[4] The larger the length, the more accurate the individual estimates.

5 Comparison to more Feature Rich Models

To place our models in the context of prior work which use a large set of linguistic features, we consider two linguistically feature rich models. Specifically, we consider a bag-of-words based Naïve Bayes' model as well as a neural network based model NEURALDATE for this task.

5.1 NAÏVEBAYES

We consider a simple, standard Multinomial Naïve Bayes classifier learned using Google Book Ngrams to date the year a document was written. We use unigram bag-of-words (we restrict our vocabulary to 200K tokens and discard out-of-vocabulary words) features and Laplace smoothing. It is worth noting that Naïve Bayes uses 200K features which is orders of magnitude higher than **NEO-Prob** approaches.

5.2 NEURALDATE

We propose a neural model NEURALDATE, to date texts. NEURALDATE operates on short sequences of words (n-grams), and outputs a probability distribution over years, $\mathsf{P}(y|\boldsymbol{x}_i)$ for each ngram $\boldsymbol{x}_i$ in the document $\boldsymbol{D}$,

Our model consists of a bi-directional LSTM with an embedding layer, two hidden layers and an output layer[5]. The embedding layer maps the input (one hot encoding of the word) to a dense embedding of size 200 dimensions. The implementation of the LSTM hidden layers are as described in (Graves and others, 2012) and therefore not described in this paper. The output layer is a soft-max layer over the years within the time-range considered. We use ADAM optimizer (Kingma and Ba, 2014) with a learning rate of $\eta = 0.001$.[6]

In order to date a document $\boldsymbol{D}$, we use the model to compute $\mathsf{P}(y|\boldsymbol{x}_i)$ for each n-gram (we use n=5) in $\boldsymbol{D}$. We then compute $\mathsf{P}(y|\boldsymbol{D})$ to be the mean of these individual probability distributions. Finally, we use the MAP estimate of $\mathsf{P}(y|\boldsymbol{D})$ as our point estimate of the year.

Autocorrelation Regularizer The model described above does not explicitly leverage structure of the label space, namely temporal structure (linear sequential structure). Observe the high variance in probability scores around the mode in Figure 4. Therefore, for a given n-gram $\boldsymbol{x}_i$ it would be preferable to learn model parameters such that $\mathsf{P}(y|\boldsymbol{x}_i)$ is "smooth" around any given label. This captures the insight that classes (years) in the neighborhood of a label l should be assigned probabilities similar to that assigned to l.[7]

We can formalize this notion of smoothness as follows: Let p_l be the probability assigned to label l. Given a neighborhood k, let $\boldsymbol{d}$ be the vector of first order differences: $p_i - p_{i+1}$ for $i \in [l-k \cdots l+k]$. We define the distribution to be L-smooth at l around neighborhood k if $\omega(\boldsymbol{d}) = \frac{\sigma(\boldsymbol{d})}{\text{mean}(|\boldsymbol{d}|)} \leq L$, for some small constant $L > 0$, where smaller values of L indicate smoother distributions.

We therefore propose to add the following cost to the original cost function:

$$\Omega(\boldsymbol{\theta}; \boldsymbol{X}^{\text{train}}) = \sum_j \omega\left(\boldsymbol{d}_j\right) ,$$

where $\boldsymbol{d}_j$ are differences between predicted probabilities for neighboring years for example j.

In summary, the final loss function including this regularization is $J(\boldsymbol{\theta}; \boldsymbol{X}^{\text{train}}, \boldsymbol{y}^{\text{train}}) + \alpha\Omega(\boldsymbol{\theta}; \boldsymbol{X}^{\text{train}})$, where $J(\boldsymbol{\theta}; \boldsymbol{X}^{\text{train}}, \boldsymbol{y}^{\text{train}})$ is the standard cross-entropy loss and α is a hyper-parameter weighing the regularization.

Figure 4 shows the effect of incorporating label smoothness constraints in the cost function for a sample n-gram. Note that incorporating the temporal structure of labels in the cost function produces markedly smoother and realistic distributions than a model not exploiting label structure.To investigate

[5]While more sophisticated and complex sequence models are being developed even as we write this paper, our goal here is only to place the performance of our proposed neologism models in context of other sophisticated methods. Therefore, Bidirectional LSTMs serve as a good lower bound for complex models.

[6]Hyper-parameter settings were chosen based on a validation set.

[7]We initially also experimented with using mean square loss and modeling this as a regression problem, but the resulting model did not perform well empirically.

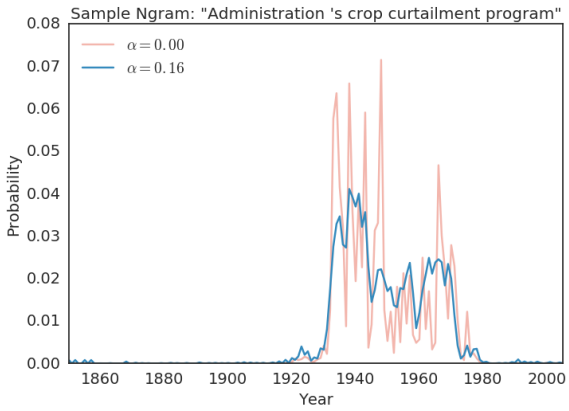


Figure 4: Predicted distribution over years for a given 5-gram (shown above the figure), motivating the need for the auto-correlation regularizer. Note that when $\alpha = 0$, the regularizer is disabled and the output probability distribution is very noisy and neighboring values have large variance. In contrast, when the regularizer is properly enabled ($\alpha = 0.16$), observe how the output probability distribution is much smoother and neighboring probability values are more similar.

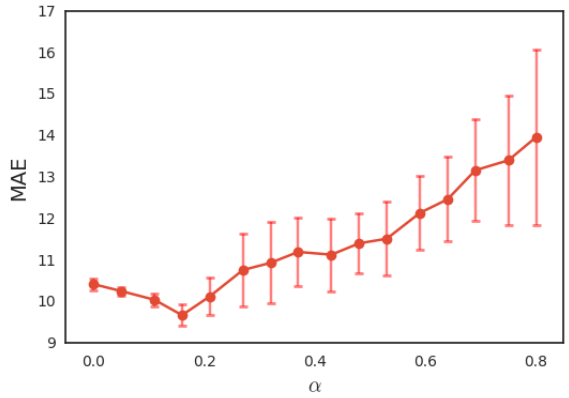


Figure 5: MAE from cross validation for candidates of α which controls the strength of the auto-correlation regularizer. The means and the standard deviations over 20 independent runs are shown. When the model is properly regularized ($\alpha = 0.16$), observe the improvements over model without regularization ($\alpha = 0$). Also, note that when the model is over-regularized ($\alpha > 0.25$), the performance is worse and demonstrates larger variance.

the effect of α, we measure the MAE (Mean Absolute Error) over n-grams and use cross-validation by selecting α from a set of candidates in $[0, 0.8]$ (see Figure 5). Based on these observations, we set α at 0.16 empirically for training our model.

6 Experiments

We evaluate all of our methods against several baselines on diverse data sets spanning multiple domains. We consider the time period of $1850 - 2005$ for the purpose of dating documents and evaluate our models on the evaluation data-sets described in Section 2. Since the tasks should get easier on long documents, we measure the performance of our models as a function of the length. Since the NYTIMES dataset only consists of the first paragraph of articles (about 100 tokens on average) we use the entire paragraph for evaluation on this dataset. Tables 2 and 3 show the Mean Absolute Error (MAE) over the NYTIMES and COHA datasets, from which we make the following observations:

- **Neologism Methods need relatively large documents to perform competitively**: Overall, the neologism based models perform competitively with Naïve Bayes using 1000 times fewer features (see

#(Tokens)	#(Features)	MAE
BOOKPROP	-	43.46
NEO	$\leq$ **200**	58.77
NEOPROBMEAN	$\leq$ **200**	27.55
NEOPROBMEDIAN	$\leq$ **200**	28.22
NEOPROBBAYES	$\leq$ **200**	67.14
NAÏVEBAYES	$\sim 200K$	23.69
NEURALDATE (w/o reg.)	1000	22.80
NEURALDATE	1000	**20.54**

Table 2: Mean Absolute Error on New York Times data. Note that neologism based methods (highlighted) whose feature set size is much smaller, perform competitively with NAÏVEBAYES with a feature space of dimension of 200K. NEURALDATE uses a 200 dimension embedding for each word in a 5-gram and so has an effective feature input size of 1000. It is worth noting that our neologism based models do not directly rely on the actual words themselves but on the number of neologisms used at a given time thus drastically reducing the feature size while yielding competitive performance.

Dataset	#(Tokens)	100	500	1000	2000
COHA-Fiction	BOOKPROP	44.57	44.54	43.95	44.21
	NEO	66.99	34.74	27.39	24.80
	NEOPROBMEAN	32.40	30.76	31.45	31.13
	NEOPROBMEDIAN	36.90	32.96	32.03	31.73
	NEOPROBBAYES	78.99	41.90	33.77	27.92
	NAÏVEBAYES	**26.61**	**23.98**	**22.62**	**21.93**
	NEURALDATE (w/o reg.)	37.56	30.71	28.96	27.97
	NEURALDATE	35.66	30.02	27.81	26.96
COHA-NonFiction	BOOKPROP	45.19	45.07	45.04	45.51
	NEO	57.99	30.75	24.84	22.86
	NEOPROBMEAN	31.13	26.90	26.02	25.39
	NEOPROBMEDIAN	30.68	26.60	25.73	25.13
	NEOPROBBAYES	56.58	30.73	25.46	21.58
	NAÏVEBAYES	**24.28**	**19.83**	**18.36**	**18.25**
	NEURALDATE (w/o reg.)	27.91	23.57	22.29	21.60
	NEURALDATE	25.21	20.07	20.38	20.09
COHA-News	BOOKPROP	44.97	45.34	44.99	45.02
	NEO	39.86	20.39	19.80	20.26
	NEOPROBMEAN	24.36	23.40	23.30	23.31
	NEOPROBMEDIAN	25.22	22.88	22.45	22.39
	NEOPROBBAYES	48.30	22.79	20.97	20.82
	NAÏVEBAYES	21.35	17.21	16.64	16.60
	NEURALDATE (w/o reg.)	20.40	16.04	15.43	15.34
	NEURALDATE	**19.30**	**15.33**	**14.72**	**14.59**

Table 3: Mean Absolute Error of different models on COHA datasets as a function of number of tokens used for evaluation in each document. Note that our proposed neologism based methods (highlighted) use a much smaller feature set, generalize across domains without any further fine-tuning, and perform competitively with feature rich models like NAÏVEBAYES and NEURALDATE for long documents (greater than 500 tokens).

Table 2) suggesting that effective usage of neologism usage patterns can serve as strong baselines.

Furthermore, from Table 3, the baseline NEO generally performs very poorly on short documents

(of length 100 tokens). For example, on the COHA-FICTION dataset using 100 tokens, the MAE is 66.99 compared to BOOKPROP which yields an MAE of 44.57. On short documents NEO is easily misled due to lack of effective sample size. In contrast, observe that as the length of the document increases NEO's error reduces significantly (note Table 3 that for 2000 word documents on COHA-FICTION the mean absolute error is now 24.80).

Finally, the probabilistic models we propose extending NEO also perform better than NEO especially on short documents (for example, on COHA-FICTION for documents with 100 tokens the MAE for NEOPROBMEAN is 32.40 compared to 66.99 for NEO). Similarly, NEOPROBMEAN and NEOPROBMEDIAN outperform NEOPROBBAYES on documents of up to 1000 tokens but NEOPROBBAYES almost always outperforms all of these on documents of length 2000, suggesting that NEOPROBBAYES needs a larger sample size to make effective predictions.

- **Deeper linguistic features boost performance**: We finally observe that including linguistic features like the words used in a simple Naïve Bayes classifier consistently outperforms methods relying solely on neologisms. Further, observe that the NEURALDATE with the auto-correlation regularizer demonstrates superior performance over NEURALDATE without regularization. Finally, note that NEURALDATE also performs competitively and sometimes out-performs Naïve Bayes.

Altogether, our proposed neologism based models generalize well across domains, reduce the input feature size significantly while performing competitively with more complex feature rich models.

7 Related Work

A large body of related work on the task of automatically dating texts exists in the field of temporal information retrieval (De Jong et al., 2005; Popescu and Strapparava, 2015; Kanhabua and Nørvåg, 2009; Garcia-Fernandez et al., 2011; Niculae et al., 2014; Zampieri et al., 2015; Zampieri et al., 2016; Bamman et al., 2017; Jatowt and others, 2017; Kumar et al., 2012; Kumar, 2013; Graliński et al., 2017). The community also has two shared tasks (Popescu and Strapparava, 2015) and (Graliński et al., 2017). However, both of these shared tasks differ from our setting. Popescu and Strapparava (2015) is a shared task for diachronic text evaluation but only focuses on the Newspaper domain in contrast to our work which focuses on generalizable models across domains. Graliński et al. (2017) is the most recent challenge on dating texts but it focuses on Polish texts. De Jong et al. (2005) proposed using temporal language models based on unigrams to date texts on Dutch newspaper articles. Several works incorporate additional features like lexical features, part-of-speech tagging, extraction of concepts and word sense disambiguation and use external knowledge bases (Kanhabua and Nørvåg, 2009; Garcia-Fernandez et al., 2011; Niculae et al., 2014; Zampieri et al., 2015; Zampieri et al., 2016). Recently Jatowt and others (2017) propose an interactive system to estimate the age of document using moment statistics of n-grams focusing on only a qualitative analysis.

Our work is most closely related to the works of (Kumar et al., 2012), (Garcia-Fernandez et al., 2011),(Zampieri et al., 2015) and (Bamman et al., 2017). Kumar et al. (2012) propose a model for predicting the dates of documents without explicit temporal cues. This model essentially learns temporal language models on the temporal corpus where explicit temporal expressions are removed. It then assesses the likelihood of a given document under each time point to make a prediction. It thus relies on implicit temporal cues and words and typically has an input feature dimensionality of the order of the vocabulary (in this case 300K words). Furthermore, this model has only been evaluated in different settings (like predicting the mid-point of an individual's lifetime using their Wikipedia biography). Garcia-Fernandez et al. (2011) develop a model to date documents using both chronological methods with external knowledge and classification methods like using an SVM to date documents on a French Newspaper corpus while Zampieri et al. (2015) propose a ranking based approach to temporal text classification. These methods learn models on the respective domains they are evaluated on. Bamman et al. (2017) proposed bag-of words based models (using Ridge Regression as well as Naïve Bayes) to predict the date of first publication over books obtained from the Hathi Trust.

Differing from these works, our proposed method seeks to learn a global model that can be applied across multiple domains without further tuning. We propose new probabilistic models to date texts by analyzing statistical patterns of the introduction of neologisms over time. Our models are simple, domain-independent, use several orders of magnitude fewer features and yet achieve competitive performance.

8 Conclusion

In this paper, we investigated the task of dating books on a large fine-grained time scale (spanning 150 years) through the lens of neologisms introduced over time. We propose probabilistic models that effectively analyze the usage of neologisms. We demonstrate that our methods perform competitively with models that use deeper linguistic cues (which could use a feature space of more than thousands of features). Furthermore, our models are learned using only the Google Book Ngrams, do not need any further tuning when evaluated on other domains and potentially enable researchers to obtain literary insights into the language of authors over time.

Acknowledgments

The authors thank the anonymous reviewers for their valuable feedback. This work was partially supported by NSF grants DBI-1355990 and IIS-1546113.

References

David Bamman, Michelle Carney, Jon Gillick, Cody Hennesy, and Vijitha Sridhar. 2017. Estimating the date of first publication in a large-scale digital library. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2017*, pages 1–10. IEEE.

Colin Chase. 1997. *The dating of Beowulf*. Number 6. University of Toronto Press.

Mark Davies. 2002. *The Corpus of Historical American English (COHA): 400 million words, 1810-2009*.

Franciska De Jong, Henning Rode, and Djoerd Hiemstra. 2005. Temporal language models for the disclosure of historical text.

Miles Efron. 2013. Query representation for cross-temporal information retrieval. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Martin Ehrensvärd. 1997. Once again: The problem of dating biblical Hebrew. *Scandinavian Journal of the Old Testament*, 11(1).

Anne Garcia-Fernandez, Anne-Laure Ligozat, Marco Dinarelli, and Delphine Bernhard. 2011. When was it written? Automatically determining publication dates. In *International Symposium on String Processing and Information Retrieval*. Springer.

Filip Graliński, Rafał Jaworski, Łukasz Borchmann, and Piotr Wierzchoń. 2017. The RetroC challenge: how to guess the publication year of a text? In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, pages 29–34. ACM.

Alex Graves et al. 2012. *Supervised sequence labeling with recurrent neural networks*, volume 385. Springer.

Avi Hurvitz. 2000. Can biblical texts be dated linguistically? Chronological perspectives in the historical study of biblical hebrew. *Vetus Testamentum-Supplements 80*, 80.

Adam Jatowt et al. 2017. Interactive system for reasoning about document age. In *CIKM*. ACM.

Adam Jatowt and Katsumi Tanaka. 2012. Large scale analysis of changes in English vocabulary over recent time. In *CIKM*. ACM.

Nattiya Kanhabua and Kjetil Nørvåg. 2009. Using temporal language models for document dating. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Abhimanu Kumar, Jason Baldridge, Matthew Lease, and Joydeep Ghosh. 2012. Dating texts without explicit temporal cues. *arXiv preprint arXiv:1211.2290*.

Abhimanu Kumar. 2013. *Supervised language models for temporal resolution of text in absence of explicit temporal cues*. Ph.D. thesis.

Jean-Baptiste Michel et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014).

Vlad Niculae, Marcos Zampieri, Liviu P Dinu, and Alina Maria Ciobanu. 2014. Temporal text ranking and automatic dating of texts. In *EACL*.

Liudmila Ostroumova Prokhorenkova, Petr Prokhorenkov, Egor Samosvat, and Pavel Serdyukov. 2016. Publication date prediction through reverse engineering of the web. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM.

Octavian Popescu and Carlo Strapparava. 2015. Semeval-2015 Task 7: Diachronic Text Evaluation. *Proceedings of SemEval*.

Mark F Rooker. 1996. Dating Isaiah 40-66: What does the linguistic evidence say?

Ian Young and Robert Rezetko. 2016. *Linguistic dating of biblical texts*, volume 1. Routledge.

Marcos Zampieri, Alina Maria Ciobanu, Vlad Niculae, and Liviu P Dinu. 2015. Ambra: A ranking approach to temporal text classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Denver, CO, USA*.

Marcos Zampieri, Shervin Malmasi, and Mark Dras. 2016. Modeling language change in historical corpora: the case of Portuguese. *arXiv preprint arXiv:1610.00030*.