Automatic Dating of Textual Documents

Rui Figueira[1], Daniel Gomes[2], and Bruno Martins[1]

[1] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
{ruimffigueira,bruno.g.martins}@tecnico.ulisboa.pt
[2] Fundação para a Computação Científica Nacional, Lisbon, Portugal
daniel.gomes@fccn.pt

Abstract. This paper addresses the automated dating of textual documents, i.e. the task of determining when a document is about or when it was written, based only on its text. We rely solely on temporal cues implicit in the text, and advance over previous work in the area by proposing a method based on a deep neural network. The proposed neural architecture explores the hierarchical nature of the input data (i.e., documents are modeled as sequences of sentences, which in turn correspond to sequences of words), combining pre-trained word embeddings, recurrent units, and neural attention, for generating intermediate representations of the textual contents. We report on experiments with data from the SemEval 2015 shared task on diachronic text evaluation, the RetroC Polish corpus designed for evaluating temporal classifiers, with documents collected from Wikipedia (e.g., biographies) and with short stories collected from project Gutenberg. Our best model corresponds to accuracy values of 35.93%,17.17%, 14.72% and 40.66%, and mean absolute error values of 36.58 , 31.86, 16.29 and 5.27 years, respectively on the SemEval, RetroC, Wikipedia and Gutenberg texts. Together, these results show that a modern neural architecture for text classification can indeed advance over previous document dating results and that, even in absence of temporal extraction resources, it is possible to achieve good results across a diverse set of texts.

1 Introduction

Temporal text mining has been, and continues to be, an active research area, both within the natural language processing (e.g., studies addressing fine-grained temporal ordering of events [7], or the recognition and contextual disambiguation of temporal expressions in textual documents [31]) and information retrieval communities (e.g., studies addressing the temporal ranking and presentation of search results [4, 17]). This paper addresses a less explored temporal text mining problem, namely the automated dating of textual documents based solely on cues implicit in the textual contents (i.e., the task of determining when a document is about or when it was written, based only on its text).

Following previous document dating work, which leveraged generative approaches based on probabilistic language models [19] or discriminative approaches

based on sparse representations and linear classifiers [26], we formulate the problem as a multi-class text classification task, discretizing the timeline into contiguous atomic time spans (i.e., the classes correspond to these discrete chronons), and attempting to infer the chronon that best corresponds to a representation of the text. However, contrarily to previous work in the area, we explore methods based on state-of-the-art deep neural network architectures.

Specifically, the proposed model combines different mechanisms for generating intermediate representations from the textual inputs, including bi-directional Gated Recurrent Units (GRUs) for modeling sequential data [6], averages of word embeddings similarly to the proposal by Joulin et al. [15], and neural attention mechanisms for highlighting relevant parts of the inputs [1]. Taking inspiration on the previous work by Yang et al. [36], the proposed network architecture explores the hierarchical nature of the input data, combining two levels of GRUs and neural attention (i.e., sentence level ad document level). The representations produced with this hierarchical approach are concatenated with a simple average of the embeddings for all the words in the input, and then passed to feed-forward nodes that output the most likely chronon.

Three output nodes are considered on the model, in an attempt to further improve results. These correspond to (i) a softmax node that outputs a year, associated to a categorical cross-entropy loss function, (ii) a softmax node that outputs the corresponding decade, also associated to a categorical cross-entropy loss, and (iii) a linear node that outputs the year, although in this case considering a loss function corresponding to the mean squared error between the predicted and ground-truth years. The entire model can be trained end-to-end from a set of labeled documents, leveraging the back-propagation algorithm in conjunction with the Adam optimization method [18].

We report on experiments with four different datasets, with different sizes and considering documents of different lengths, and considering also different temporal periods. These include (a) data from the SemEval 2015 shared task on diachronic text evaluation [28], (b) the RetroC Polish corpus designed for evaluating temporal classifiers [11], (c) documents collected for Wikipedia (e.g., biographies) [14] and (d) short stories collected from project Gutenberg. The obtained results attest to the robustness of the proposed approach across a diverse set of prediction tasks (e.g., for document collections spanning hundreds and thousands of years, or much shorter temporal periods), confirming the usefulness of the implicit temporal cues available in general textual contents. Our full model corresponds to accuracy values of 35.93%,17.17%, 14.72% and 40.66%, and mean absolute error values of 36.58, 31.86, 16.29 and 5.27 years, respectively on the SemEval, RetroC, Wikipedia and Gutenberg texts. Together, these results show that a modern neural architecture for text classification can indeed outperform some previous methods and that, even in the absence of temporal extraction resources, it is possible to achieve good results across a diverse set of texts.

The rest of this paper is organized as follows: Section 2 surveys previous related work. Section 3 details the proposed approach, presenting the architecture

of the deep neural network that was considered for addressing document dating as a supervised classification task. Section 4 presents the experimental evaluation of the proposed method, detailing the datasets, the evaluation methodology, and the obtained results. Finally, Section 5 summarizes our main conclusions and presents possible directions for future work.

2 Related Work

Temporal text mining is a widely explored area within natural language processing and information retrieval, although relatively few studies have addressed document dating. In the 2015 edition of the International Workshop on Semantic Evaluation (SemEval 2015), Task 7 addressed this issue with a diachronic text evaluation challenge. The task was composed by 3 different subtasks: The first one considered pieces of news in which specific historical events or named entities are clearly mentioned, while in the second one such information is missing, but there are enough clues to assign a temporal interval to each document. The third one is quite different from the previous tasks and consisted of handling phrases in context (i.e., phrases with specific expressions from a determined historical epoch). In the first two subtasks, each textual sample got 3 types of intervals with the following granularity: Fine (2 years interval for Subtask 1, and 6 years for Subtask 2), Medium (6 years interval for Subtask 1, and 12 years for Subtask 2) and Coarse (12 years interval for Subtask 1, and 20 years for Subtask 2). A total of 3 teams participated on the shared task [28].

USAAR [33] was the best performed method in Subtask 1. This method is basically a web crawler that uses the snippets present in the texts for a web search. The date retrieved is assigned to the text sample. With this method they obtained 98.1% precision in the coarse granularity. IXA [30] was the only proposal to submit results to the 3 subtasks. Their general method takes into account four approaches in order to determine the time period of time in which the piece of news was written. The first approach consists of searching for time mentions in the piece of text; the second one consists of searching for named entities in the text, and then linking them with the time period described in Wikipedia; the third approach used Google NGrams[1] to calculate the year probability associated to each noun present in the text. Finally, the fourth approach consists of using relevant linguistic features to language change, in combination with machine learning. With this last model, the authors managed to achieve, in the coarse granularity, 9% precision in Subtask 1 and 9.8% precision in Subtask 2. In subtask 2, the best performed approach was UCD [32]. With an SVM classifier for each granularity (i.e., 6, 12 or 20 years) and using a set of stylistic features from the texts, such as frequency counts of words and characters combined with the Google Books Syntactic N-Grams (GSN) database, the authors managed to achieve a precision of 54.2% for the coarse granularity. AMBR [37],

[1] `https://books.google.com/ngrams`

the final approach presented, is an approach based on a learning-to-rank framework using pairwise comparison [22]. Their classifier is trained to learn a linear function which preserves the temporal ordering of the documents. They compare each new training example with the ones already in the dataset deciding, one by one, which is older and which is newer, constructing a dataset ordered like a timeline. To make a prediction, the score of the new example is calculated using the trained linear function, and then the predicted interval is the one minimizing the average distance to the k-nearest neighbours. With this approach the authors managed to achieve, in the coarse granularity, 7.4% precision in Subtask 1 and 29.2% precision in Subtask 2.

Another automatic temporal dating challenge was proposed by Graliński et al. [10] using their RetroC corpus of historical Polish texts [11]. From all the presented proposals, the best performing system consisted in a combination of a basic neural networks with a regression model built using the Vowpal Wabbit open-source learning system [20]. As features the athors used lower case tokes and/or pentagrams. The weights are atributed to the whole document on the base of token frequencies within years. The authors managed to achieve a RMSE of 24.8 years in the test set and 17.2 yeas in the development set. Another proposal, consisting in a pure neural network, presented very similar results (RMSE of 24.9 years), but accordingly to Graliński et al. the experiment was not fully transparent .

Kumar et Al. [19] implemented a simple method based on the concept of maximum likelihood estimation. With a corpus composed by short stories collected from the Gutenberg Project website, and leveraging the information present on Wikipedia biographies, the authors proposed a method based on the creation of a language model for each *chronon*, and then measuring the maximum likelihood between the test document and each language model. With this approach they achieved a MAE of 34 years.

In another study presented in the automatic dating area, Jatowt et al. [14] presented a graph-based method to date textual samples. The basic idea of their approach is to generate a co-occurrence graph reflecting the relations between words and dates, and then using the generated graph to attribute the most suitable date to each test document. To create the graph, the authors formed 5 news article collections with documents collected from the Google News Archive[2]. Each of this collections represents one different country (i.e., Germany, UK, Japan, France and Israel). These 5 collections, with an average of 100k news articles each, were used to create 5 different co-occurrence graphs. Based on each collection, the authors constructed a graph $G(V, E)$ where V is the set of vertices, with each vertice representing a unique word, while E is the set of relationships between words. The creation of the co-occurrence graph is divided into two stages: the calculation of word-time associations and the estimation

[2] `http://news.google.com/archivesearch`

of the temporal weights. The graph is created based on the co-occurrence of words and dates (that are also treated as words), and also in the premise that, if word w strongly co-occurs with many other words associated to a time point t, so the word w should also be associated with the time point t. The temporal weights of each association are trained using the Google News Archive dataset. A more detailed review of the model can be found on the main article of my M.Sc. thesis. The authors end up proposing two methods: The first one the base graph method with no special attention to any specific word, and an extended version of the same method that gives special attention to the dates present in the documents. To test their models the authors prepared 3 different datasets: A Wikipedia dataset with 250 major historical events from the 5 different countries. The second dataset was composed by excerpts from two historical books. The last one was composed by excerpts from news websites. Regarding to the most relevant dataset to this work, the Wikipedia dataset, the authors managed to achieve an average error of 2.84 years in the extended version, while the basic model produced an average error of 18.3 years.

3 Proposed Methodology

Taking inspiration on previous work addressing text classification tasks [8, 15, 36], we propose a neural network model for determining when a document is about or when it was written, based only on its text. The document dating task is formulated as a multi-class text classification problem, discretizing the timeline into contiguous atomic time spans (i.e., the classes correspond to these discrete chronons, each with a duration of one year), and attempting to infer the chronon that best corresponds to a representation of the text. Figure 1 presents the proposed neural network architecture, which is detailed next. For an in-depth introduction to deep neural networks for natural language processing, complementing the descriptions from the following sections, the reader can refer to the tutorial by Goldberg [9].

Noting that the inputs to our model can be seen as having a hierarchical structure (i.e., sequences of words form the different sentences, and each document is composed of a sequence of sentences), our model first builds representations of individual sentences, and then aggregates those into an encompassing representation. This two-level hierarchical approach is illustrated in Figure 1, with the word-level part of the model (i.e., the part that generates a representation from a given sentence, based on the composing words) shown in the box at the top. A recurrent neural network node known as a Gated Recurrent Unit (GRU) is used at both levels to build the representations (shown in purple in the word-level, and in blue in the sentence-level), and we specifically considered bi-directional GRUs [6] combined with neural attention mechanisms [36]. Notice that the GRUs in the first level of the model leverage word embeddings as input, whereas the second level uses as input the sentence representations (shown in green) generated at the first level. In the case of texts in the English language,

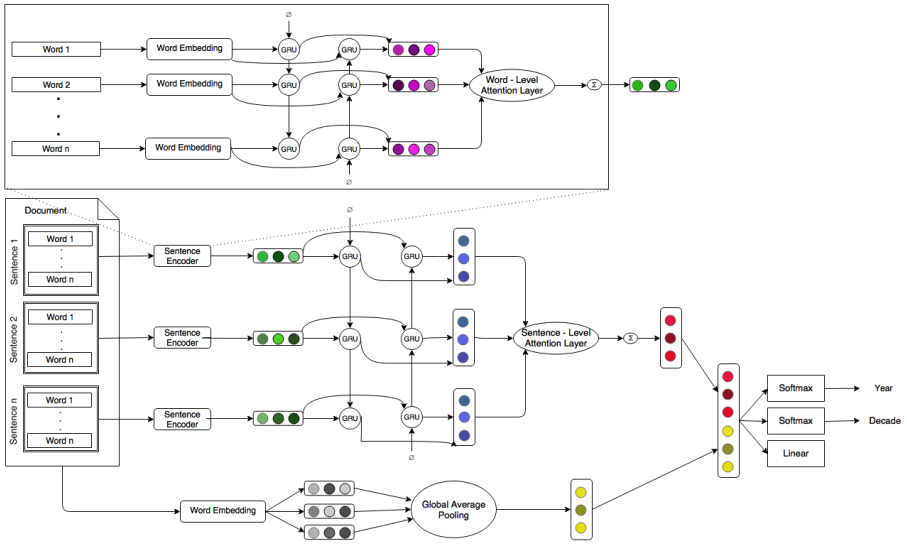

Fig. 1. Neural Network Architecture Schema

we specifically leveraged 300-dimensional word embeddings, pre-trained on the Common Crawl and considering cased words, made available in the context of the GloVe project [27]. The experiments with texts in Polish language were similarly made, with the use of pre-trained word embeddings made available by Bojanowski et al. [2]. The embeddings layer is initialized with basis on these pre-trained values, and then adjusted during model training.

GRUs model sequential data by having a recurrent hidden state whose activation at each time step is dependent on that of the previous time step. A GRU computes the next hidden state h_t given a previous hidden state h_{t-1} and the current input x_t using two gates (i.e., a reset gate r_t and an update gate z_t), that control how the information is updated, as shown in Equation 1. The update gate (Equation 2) determines how much past information is kept and how much new information is added, while the reset gate (Equation 4) is responsible for how much the past state contributes to the candidate state. In Equations 1 to 4, $\tilde{h}_t$ stands for the current new state, W is the parameter matrix for the actual state, U is the parameter matrix for the previous state, and b a bias vector.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{1}$$

$$z_t = \sigma\big(W_z x_t + U_z h_{t-1} + b_z\big) \tag{2}$$

$$\tilde{h}_t = \tanh\big(W_h x_t + r_t \odot (U_h h_{t-1} + b_h)\big) \tag{3}$$

$$r_t = \sigma\big(W_r x_t + U_r h_{t-1} + b_r\big) \tag{4}$$

Bi-directional GRUs perceive the context of each input in a sequence by outlining the information from both directions. Concatenating the output of processing a sequence forward $\overrightarrow{h}_{it}$ and backwards $\overleftarrow{h}_{it}$ grants a summary of the information around each position, $h_{it} = [\overrightarrow{h}_{it}, \overleftarrow{h}_{it}]$.

Since words and sentences can be differently informative in specific contexts, the model also includes two levels of attention mechanisms (i.e., one at the word level and one at the sentence level), that let the model to pay more or less attention to individual words/sentences when constructing representations. For instance, in the case of the word-level part of the network, the outputs h_{it} of the bi-directional GRU encoder are fed to a feed-forward node (Equation 5), resulting in vectors u_{it} representing words in the input. A normalized importance α_{it} (i.e., the attention weights) is calculated as shown in Equation 6, using a context vector u_w that is randomly initialized. The importance weights in α_{it} are then summed over the whole sequence, as shown in Equation 7.

$$u_{it} = \tanh\left(W_w h_{it} + b_w\right) \tag{5}$$

$$\alpha_{it} = \frac{\exp\left(u_{it}^T u_w\right)}{\sum_t \exp\left(u_{it}^T u_w\right)} \tag{6}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{7}$$

The vector s_i from Equation 7 is finally taken as the representation of the input. The part of the network that processes the sequence of sentences similarly makes use of bi-directional GRUs with an attention mechanism, taking as input the representations produced for each sentence, as shown in Figure 1.

The representation that is produced as the output of the sentence-level attention mechanism (represented in red), which encompasses the entire output, is also concatenated with an alternative representation built through a simpler mechanism which, taking inspiration on the good results reported by Joulin et al. [15], computes the average of the embeddings for all words in the input sentences (shown in yellow). The word embeddings layer is shared by the hierarchical attention and the averaging mechanisms, and thus while one part of the model uses multiple parameters to compute representations for the inputs, the other part of the model acts as a shortcut that can more directly propagate errors back into the embeddings, so that they can be updated.

The output layer of the model considered 3 different nodes, each one with a different output value. The first one consists in a softmax node associated to a categorical cross-entropy loss function, producing an output label corresponding to the most suitable year to each test sample. This can be considered as the main output node since all the metrics presented are calculated considering the output of this node. The main intention of the other two output nodes was to improve the global performance of the model. The second node is also a softmax and also associated with a categorical cross-entropy loss function, but in this case producing an output corresponding to the predicted decade of each test sample. The final output layer is a linear node which considers a loss function

corresponding to the mean squared error between the predicted year and the ground-truth year.

The entire model is trained end-to-end from a set of documents assigned to gold-standard years, leveraging the back-propagation algorithm [29] in conjunction with the Adam optimization method [18]. In the combined loss function, the three output nodes have weights of 1.0, 1.0 and 0.25, respectively. The word embeddings layer considered a dimensionality of 300, and the output of the GRUs had a dimensionality of 150. Model training was made in batches of 32 instances, using the default parameters for the Adam optimization algorithm, and considered a stopping criteria based on the combined training loss, finishing when the difference between epochs was less than 0.0000001 with a patience value of 2. The implementation of the model relied mostly on the keras[3] deep learning library, although the scikit-learn[4] machine learning package was also used for specific operations (e.g., for computing the evaluation metrics and for creating the cross-validation data splits that were considered in the experimental evaluation).

4 Dataset and Experimental Results

The experimental validation of the proposed method leveraged a total of 4 different datasets.

The first dataset taken into account in our experiments is the one from Subtask 2 of the SemEval 2015 Task 7. The dataset is composed by 4168 training examples and 1041 test samples, with their distributions shown in Figure 2. The dataset is composed by parts of text from various news, present in journals available in electronic format, specially NPA[5], SPR[6] and BDY[7]. Unlike in Subtask 1, were there are historical events or named entities clearly mentioned, in Subtask 2 such information is missing, but there are enough clues to assign a time interval to the excerpt, at least for a human being. Each of the instances has a temporal annotation according to 3 types of intervals: Fine (6 years), Medium, (12 years) and Coarse (20 years). We processed the dataset, representing each interval by his middle point. This means that, if our method's prediction is equal to the correct class (i.e., the interval middle point), our prediction is correct.

In the SemEval 2015, there were 3 proposals presented to this subtask 2, as described in the Section 2. Table 2 presentes the precision results obtained in the coarse granularity, as well as the mean absolute error in years for the two best proposals.

The second dataset was collected by us from the Australian Gutenberg project website[8]. The Gutenberg Project is a digital collection of eBooks, which

[3] http://keras.io
[4] http://scikit-learn.org
[5] http://newspaper.archive.com
[6] http://archive.spectator.co.uk/
[7] http://www.bodley.ox.ac.uk/ilej/
[8] http://gutenberg.net.au/

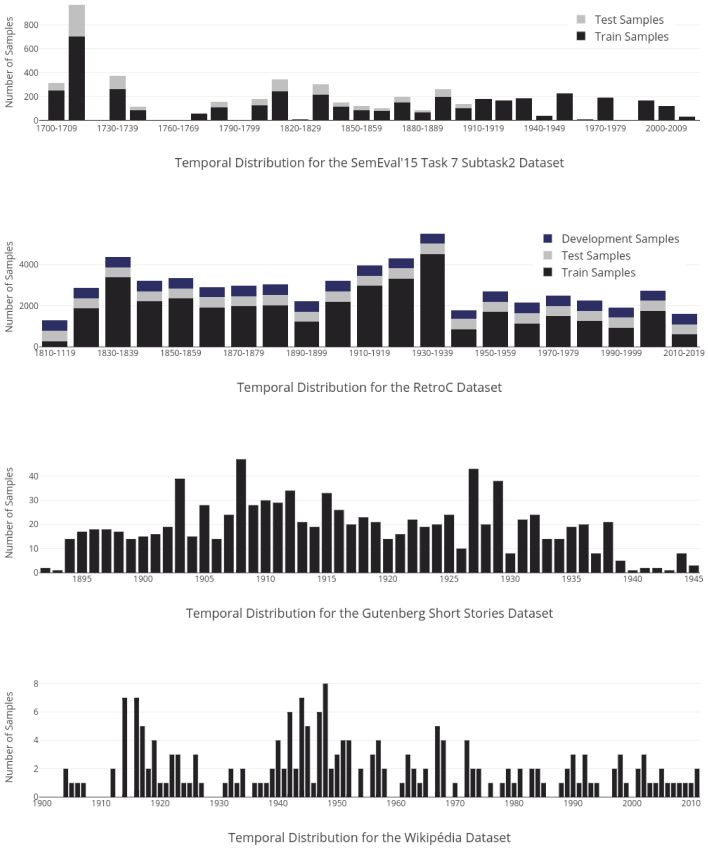

Fig. 2. Distributions of the Considered Datasets

relies on a community effort to digitize and share out-of-copyright works. The dataset, composed by short stories with publication dates between 1892 and 1945, has a total of 1000 instances that are temporal distributed as shown in Figure 2. Kumar et al. [19] in their work detailed in the Section 2 also used a collection of short stories to validate their work. However their methodology was a little bit differen for the one followed by us, since Kumar et al. used an external corpus of biographies from Wikipedia to tune the parameters.

Wikipedia pages are also valuable for evaluation purposes. This free online encyclopedia is based on collaborative writing by its users, and events described in Wikipedia usually have their focus time well defined. We used a dataset based in the one that Jatowt et al. used in their previous work [14]. It is composed by 195 historical events from 4 different countries with the following distribution: 48 events from UK and France, 49 events from Germany and 50 events Israel. The initial dataset from Jatowt et al. was composed by 250 samples (50 from

Table 1. Statistical characterization for the datasets used in our experiments.

	SemEval	RetroC	Gutenberg	Wikipedia
Number of documents (train/test)	4168/1041	40000/9910	1000	195
Number of classes (years)	311	200	54	108
Vocabulary size	26041	1738577	88650	55744
Number of words p/doc (avg.)	66.331	513.729	4711.837	346.441
Number of sentences p/doc (avg.)	2.478	39.974	336.101	9.138

Table 2. Summary of previous results

	Best Result				Interesting Alternative			
	Reference	MAE	RMSE	Accuracy	Reference	MAE	RMSE	Accuracy
SemEval	Szymanski et al. [32]	19	—	0.542	Zampieri et al. [37]	31.74	—	0.292
RetroC	VW + NN by Graliński et al. [10]	—	17.2	—	VW only by Graliński et al. [10]	—	22.0	—
Gutenberg	Kumar et. al [19]	34	—	—	—	—	—	—
Wikipedia	Extended Proposal by Jatowt et al. [14]	2.83	—	—	Basic Proposal by Jatowt et al. [14]	18.3	—	—

each country), but since the study was published, some of the Wikipedia pages were removed and/or merged. We were also unable to collect the 50 events from Japan. All the events selected occurred between 1900 and 2013.

The ground truth date associated to the Wikipedia document was set to the middle point between the start and end date of each event (i.e., with the same procedure used in the SemEval dataset). The temporal distribution of the full dataset can be seen in the Figure 2.

The final datasets used to validate our model was *RetroC* [11]. This dataset is a Polish-language diachronic corpus, mostly based on publications available in Polish digital Libraries, and with publication times between 1814 and 2013. This dataset is composed by 59910 samples divided as shown in Table 1. We used the training sample to train and the development samples to test the model because the train samples made available publicly do not have the label attached, so we could not assess the results.

From the 4 datasets previously cited and used to validate, two of them were already splitten into train and validation splits, namely the SemEval and RetroC ones. In this cases we use this splits in our test in order to compare our results to the obtained in the competitions. In the other 2 cases we use cross validation to obtain our results. We use 10 folds for the Gutenberg (due to the high number of samples) and 5 folds for the Wikipedia dataset (due to the smaller number of samples).

The experimental results can be summarized by various statistics, such as the predictive accuracy (i.e., the percentage of documents assigned to the correct year), Root Mean Square Error (RMSE) between the estimated and the ground-truth dates, or the Mean Absolute Error (MAE). The formulas corresponding to these last two metrics are as follows.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{8}$$

$$MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \tag{9}$$

In Equations 8 and 9, $\hat{y}_i$ corresponds to a predicted date, y_i corresponds to a true date, and n is the number of predictions. Using multiple error metrics can have advantages, given that individual measures condense a large number of data into a single value, thus only providing one projection of the model errors that emphasizes a certain aspect of model performance. For instance, Willmott and Matsura [35] proved that the RMSE is not equivalent to the MAE, and that one cannot easily derive the MAE value from the RMSE (and vice versa). While the MAE gives the same weight to all errors, the RMSE penalizes variance, as it gives errors with larger absolute values more weight than errors with smaller absolute values. When both metrics are calculated, the RMSE is by definition never smaller than the MAE. Chai and Draxler [5] argued that the MAE is suitable to describe uniformly distributed errors, but because model errors are likely to have a normal distribution rather than a uniform distribution, the RMSE is often a better metric to present than the MAE. Multiple metrics can provide a better picture of error distribution and thus, in our study, we present results in terms of the MAE and RMSE metrics.

In our experiments we use the percentile 75 values for the number of sentences per text and words per sentence values. With this values we guarantee a small lose of information in the 25% bigger samples, while avoiding the waste of information in the smaller ones. Table 3 sum up the percentile 75 values for the 4 datasets used in our experiments.

In Table 4 we have summarized the results obtained with our model in the 4 different datasets.

We performed 3 different tests with our model. The first two columns of the table present the two simpler approaches, considering each one of them one of the two branches of our model. The first one considers only the inputs received (i.e., of the words that composes the sentences of the texts) and makes an average of the embeddings associated to the inputs. This can be considered the simplest branch of our model. The second one presents the performance of the hierarchical attention mechanism. In this branch the input is also the embedding layer, but then are applied the Bidirectional GRUs and attention layers at the word and sentence levels. The Full Model approach is the one that concatenates the two previously explained branches.

Note that in the Wikipedia results, the accuracy result presented measures the number of correct predictions considering their real time interval, while the value in parenthesis only considers the middle point accuracy.

In the SemEval dataset we managed to achieve an accuracy of 35.93%. This result would rank our solution in the second place only outscored by the UCD [32] proposal which had 54.2%. This improvement in the results of the UCD proposal can possibly be explained by the usage of the GNS database.

The results on the RetroC dataset were not impressive. We were not able to approach the best performed methodology (i.e., the vowpall wabbit regression combined with the neural network). In any case, with a RMSE of 31.86 years in the full model, our approach performed way better than their all of the base models presented by Graliński et al. [10].

Table 3. Percentile 75 for different the Datasets

	SemEval	RetroC	Gutenberg	Wikipedia
Number of Sentences	3.0	47.0	449.0	10.0
Words per Sentences	37.0	18.0	19.0	47.0

Table 4. Summary of the Model Results

	Avg Embeddings			Hierarchical Attention			Full Model		
	Accuracy	MAE	RMSE	Accuracy	MAE	RMSE	Accuracy	MAE	RMSE
SemEval	35.06%	35.49	64.35	7.20%	73.40	98.15	35.93%	36.58	63.90
RetroC	18.03%	15.61	30.16	0.05%	52.50	61.54	17.17%	16.47	31.86
Gutenberg	44.64%	4.35	8.53	5.72%	11.16	14.21	40.66%	5.27	9.57
Wikipedia	16.61% (15.27%)	10.06	28.7	12.22% (9.33%)	18.09	26.83	14.72% (14.28%)	16.29	24.52

In the Gutenberg dataset, ours results were way better than the ones presented by Kumar et. al [19]. We were able to achieve a MAE of 5.27 years in the full model, which is way lower than the 34 years presented in Kumar's paper. In any case, we should regard that we used a different dataset, since the dataset used in their work was not publicly available.

The results with the Wikipedia dataset were very close to the ones achieved by Jatowt et al. [14]. Even taking into account slight differences in the dataset (already stated before) and in the training process, and considering their basic model, we managed to improve their result of 18.3 years to 16.29 years. Note that we, in our proposal, don't give special attention to the dates present in the documents, so our approach were unable to outscore their extended method.

The attention mechanism, as implemented in our model, can also offer model interpretability, allowing us to see which parts are more or less relevant to the classification task. In Figure 3 we can see two examples of attention weights assigned to the sentences and words in two different textual documents from the SemEval collection.

In the first document, we can see that the sentence 1 is the one with lower attention value. In the sentence with highest weight (i.e., sentence 2) we can see that there is a year reference (i.e., 1808). The words *Since* and *1808* are stated as two of the most relevant words in the sentence. These values shows that the model gives more relevance to dates instead of more common words.

In the second example, we can see that Sentence 2 is much more relevant than Sentence 1. In the presented case, Sentence 2 got an attention value of 0.55, while Sentence 1 only has an attention value of 0.12. One of the possible reasons for that is the presence of the term *ere*, which is archaic word for *before*, mostly used in the XIX century. In fact, the ground truth value for this document is 1836, which means that this particular term can have an high impact in the prediction results for this example.

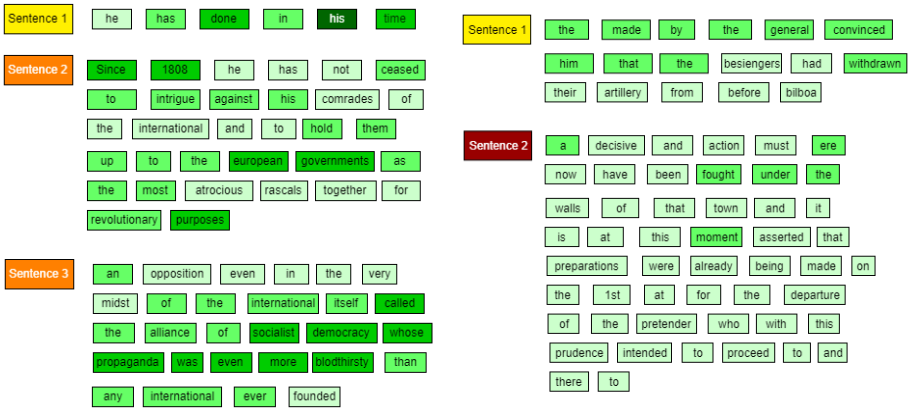

Fig. 3. Examples of the Distributions of the Attention Weights for two Documents

5 Conclusions and Future Work

Temporal text mining is an area with growing interest in the NLP community, and with this raising popularity, the number of studies and approaches presented also increases. This novel automatic dating method combines two approaches (never explored in the temporal resolution area), trying to perceive and leverage the context of the words and sentences in textual samples.

With this new proposal we introduce a new method with the usage of an attention mechanism, which were never been explored in any previous automatic dating studies. This model, leveraging the hierarchical structure of any textual document, tries to perceive the context and meaning of words and sentences enhancing the results obtained with a gradual adjustment of the attention weights attributed to each word and sentence.

All in all, the results obtained were not perfect but in some cases we were able to approach or surpass some of the state-of-the-art results. Additionally, we prove that with the presence of the attention mechanism we were able to achieve model interpretability, which means that we are able to assess which words and/or sentences are more relevant in each prediction.

Despite the results obtained there are also many ideas for future work, since different options can be considered for improving the neural network architecture. For instance, to better handle out-of-vocabulary words and issues related to changes in word spelling over time, we could consider alternative mechanisms for exploring context in the generation of the word embeddings, or replacing/enriching the embeddings with mechanisms that generate representations from individual characters or character n-grams [3].

Other parts of the neural model architecture can also be changed. Our neural architecture leverages GRUs to encode sequences of words, but other types of re-

current nodes have also recently been proposed. For instance, the Minimal Gated Unit approach [13,38] relies on a simplified model with just a single gate. Having less parameters to train can contribute to improving the model effectiveness. In contrast, Multi-Function Recurrent Units (Mu-FuRUs) adopt an elaborate gating mechanism that allows for additional differentiable functions as composition operations, leading to models that can better capture the nuances involved in encoding word sequences [34]. Other alternatives include Long Short-Term Memory (LSTM) networks with coupled gates [12], Structurally Constrained Recurrent Networks [24], IRNNs [21], and many other LSTM or GRU variants [12, 16].

Another idea yet relates to the use of sparse modeling methods as an approach to improve the predictions at the output nodes, by using sparsemax instead of the softmax activation at the model outputs [23]. Sparse modeling methods could also be used as an approach to improve the interpretability of the attention mechanisms [25] (i.e., standard attention tends to produce dense outputs, in the sense that all elements in the input always make at least a small contribution to the decision, while sparse alternatives can better encourage parsimony and interpretability).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations (2014)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5 (2017)
4. Campos, R., Dias, G., Jorge, A.M., Jatowt, A.: Survey of temporal information retrieval and related applications. ACM Computing Surveys 47(2) (2014)
5. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature. Geoscientific Model Development 7(3) (2014)
6. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of the Workshop on Synthax, Semantics and Structure in Statistical Translation (2014)
7. Derczynski, L.R.: Automatically Ordering Events and Times in Text. Springer International Publishing (2017)
8. Duarte, F., Martins, B., Sousa Pinto, C., Silva, M.: A deep learning method for icd-10 coding of free-text death certificates. In: EPIA 2017:Progress in Artificial Intelligence. (2017)
9. Goldberg, Y.: A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research 57 (2016)
10. Graliński, F., Jaworski, R., Borchmann, L., Wierzchoń, P.: The RetroC challenge: How to guess the publication year of a text? In: International Conference on Digital Access to Textual Cultural Heritage (2017)
11. Graliński, F., Wierzchon, P.: RetroC – A Corpus for Evaluating Temporal Classifiers. In: Proceedings of Language & Technology Conference (2015)

12. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems (2016)
13. Heck, J., Salem, F.M.: Simplified minimal gated unit variations for recurrent neural networks. arXiv preprint arXiv:1701.03452 (2017)
14. Jatowt, A., Au Yeung, C.M., Tanaka, K.: Estimating document focus time. In: Proceedings of the ACM International Conference on Information & Knowledge Management (2013)
15. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. Computing Research Repository abs/1607.01759 (2016)
16. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: Proceedings of the International Conference on Machine Learning (2015)
17. Kanhabua, N., Blanco, R., Nørvåg, K.: Temporal Information Retrieval. Foundations and Trends in Information Retrieval, Now Publishers Inc (2015)
18. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference for Learning Representations (2015)
19. Kumar, A., Lease, M., Baldridge, J.: Supervised language modeling for temporal resolution of texts. In: Proceedings of the ACM International Conference on Information & Knowledge Management (2011)
20. Langford, J., Li, L., Zhang, T.: Sparse online learning via truncated gradient. Journal of Machine Learning Research 10(Mar) (2009)
21. Le, Q.V., Jaitly, N., Hinton, G.E.: A simple way to initialize recurrent networks of rectified linear units. arXiv preprint arXiv:1504.00941 (2015)
22. Liu, T.Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval (2009)
23. Martins, A.F.T., Astudillo, R.F.: From softmax to sparsemax: A sparse model of attention and multi-label classification. In: Proceedings of the International Conference on Machine Learning (2016)
24. Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., Ranzato, M.: Learning longer memory in recurrent neural networks. arXiv preprint arXiv:1412.7753 (2014)
25. Niculae, V., Blondel, M.: A Regularized Framework for Sparse and Structured Neural Attention. arXiv preprint arXiv:1705.07704 (2017)
26. Niculae, V., Zampieri, M., Dinu, L.P., Ciobanu, A.M.: Temporal text ranking and automatic dating of texts. Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (2014)
27. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods on Natural Language Processing (2014)
28. Popescu, O., Strapparava, C.: Semeval 2015, Task 7: Diachronic text evaluation. In: Proceedings of the International Workshop on Semantic Evaluation (2015)
29. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Cognitive Modeling 5(3) (1988)
30. Salaberri, H., Salaberri, I., Arregi, O., Zapirain, B.: Ixagroupehudiac: A multiple approach system towards the diachronic evaluation of texts. In: Proceedings of the International Workshop on Semantic Evaluation (2015)
31. Strötgen, J., Gertz, M.: Domain-sensitive temporal tagging. Morgan & Claypool Publishers, Cham, Switzerland (2016)
32. Szymanski, T., Lynch, G.: UCD: Diachronic text classification with character, word, and syntactic n-grams. In: Proceedings of the 9th International Workshop on Semantic Evaluation (2015)

33. Tan, L., Ordan, N., et al.: Usaar-chronos: Crawling the web for temporal annotations. In: Proceedings of the 9th International Workshop on Semantic Evaluation (2015)
34. Weissenborn, D., Rocktäschel, T.: Mu-FuRU: The multi-function recurrent unit. In: Proceedings of the Association for Computational Linguistics Workshop on Representation Learning for NLP (2016)
35. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research 30(1) (2005)
36. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (2016)
37. Zampieri, M., Ciobanu, A.M., Niculae, V., Dinu, L.P.: AMBRA: A ranking approach to temporal text classification. In: Proceedings of the 9th International Workshop on Semantic Evaluation (2015)
38. Zhou, G.B., Wu, J., Zhang, C.L., Zhou, Z.H.: Minimal gated unit for recurrent neural networks. International Journal of Automation and Computing 13(3) (2016)