# When Was It Written? Automatically Determining Publication Dates

Anne Garcia-Fernandez[1,⋆], Anne-Laure Ligozat[1,2], Marco Dinarelli[1], and Delphine Bernhard[1]

[1] LIMSI-CNRS, Orsay, France
[2] ENSIIE, Evry, France
{annegf,annlor,marcod,bernhard}@limsi.fr

**Abstract.** Automatically determining the publication date of a document is a complex task, since a document may contain only few intra-textual hints about its publication date. Yet, it has many important applications. Indeed, the amount of digitized historical documents is constantly increasing, but their publication dates are not always properly identified via OCR acquisition. Accurate knowledge about publication dates is crucial for many applications, e.g. studying the evolution of documents topics over a certain period of time.

In this article, we present a method for automatically determining the publication dates of documents, which was evaluated on a French newspaper corpus in the context of the DEFT 2011 evaluation campaign. Our system is based on a combination of different individual systems, relying both on supervised and unsupervised learning, and uses several external resources, e.g. Wikipedia, Google Books Ngrams, and etymological background knowledge about the French language. Our system detects the correct year of publication in 10% of the cases for 300-word excerpts and in 14% of the cases for 500-word excerpts, which is very promising given the complexity of the task.

## 1 Introduction

Automatically determining the publication date of a document is a complex task, since a document may contain only few intra-textual hints about its publication date. This task has many important applications including temporal text-containment search [13] and management of digitized historical documents. Indeed, the amount of digitized historical documents is constantly increasing, but their publication dates are not always properly identified by automatic methods.

In this article, we present a novel method for automatically determining the publication dates of documents, which was evaluated on a French newspaper corpus in the context of the DEFT 2011[1] evaluation campaign [5]. Our approach combines a large variety of techniques, based on both a training corpus and

⋆ The author is now working at CEA-LIST, DIASI-LVIC lab at Fontenay-Aux-Roses, France.

[1] http://deft2011.limsi.fr/

R. Grossi et al. (Eds.): SPIRE 2011, LNCS 7024, pp. 221–236, 2011.

external resources, as well as supervised and unsupervised methods. The main contributions of the paper are as follows:

- We use the Google Books Ngrams, which were made recently available by Google, in order to automatically identify neologisms and archaisms.
- We build classification models on a corpus covering a large range of historical documents and publication dates.
- We apply Natural Language Processing techniques on challenging OCRized data.
- We study and evaluate different independent systems for determining publication dates, as well as several combination techniques.

In the next section, we discuss the state of the art. In section 3 we detail the training and evaluation corpora as well as the evaluation methodology. In section 4 we describe corpus independent approaches, which we call "chronological methods", while in section 5 we describe supervised classification methods. Combination techniques for aggregating the individual systems are detailed in section 6. Finally, we evaluate the systems in section 7 and conclude in section 8 providing some perspectives for future work.

## 2 State of the Art

Though there is an extensive literature on text categorization tasks, research on temporal classification is scarce. Existing approaches are based on the intuition that, for a given document, it is possible to find its publication date by selecting the time partition whose term usage has the largest overlap with the document. The models thus assign a probability to a document according to word statistics over time.

De Jong et al. [3] aim at linking contemporary search terms to their historical equivalents and at dating texts, in order to improve the retrieval of historical texts. They propose building independent language models for documents and time partitions (with varying granularities for model and output), using unigram models only. Then the divergence between the models of a partition and a tested document is measured by a normalized log-likelihood ratio with smoothing. Due to the lack of huge digitized reference corpora, the experiments are performed on contemporary content only, consisting of articles from Dutch newspapers, with a time span ranging from 1999 to 2005. The models based on documents outperform those based on time partitions.

Kanhabua and Nørvåg [8] reuse the previous model, but incorporate several preprocessing techniques: part-of-speech tagging, collocation extraction (e.g. "United States"), word sense disambiguation, concept extraction and word filtering (tf-idf weighting and selection of top-ranked terms). They also propose three methods for improving the similarity between models: word interpolation (smoothing of frequencies to compensate for the limited size of corpora), temporal entropy (to measure how well a term is suited for separating a document from other documents in a document collection) and external search statistics

from Google Zeitgeist (trends of search terms). They created a corpus of about 9,000 English web pages, mostly web versions of newspapers, covering on average 8 years for each source. The techniques were evaluated for time granularities ranging from one week to one year. The preprocessing techniques improved the results obtained by de Jong et al. [3]. This work lead to the creation of a tool for determining the timestamp of a non-timestamped document [9].

The DEFT 2010 challenge proposed a task whose goal was to identify the decade of publication of a newspaper excerpt [6]. The corpus was composed of articles from five French newspapers, automatically digitized with OCR (Optical Character Recognition) and covering a time range of a century and a half. The best performing system [1] obtained an f-measure of 0.338 using spelling reforms, birth dates, and learning of the vocabulary. The second best system [15] used orthographic correction, named entity recognition, correction with Google Suggest, date search on Wikipedia, and language models.

## 3 Methodology

### 3.1 Corpus Description

The dataset used for training and evaluating our system was provided in the context of the DEFT 2011 challenge.

The corpora were collected from seven French newspapers available in Gallica:[2] *La Croix, Le Figaro, Le Journal de l'Empire, Le Journal des Débats, Le Journal des Débats politiques et littéraires,* and *Le Temps* plus an unknown newspaper present only in the evaluation data set. The corpus is composed of article excerpts, called *portions*, containing either 300 or 500 words and published between 1801 and 1944. The excerpts with 300 or 500 words were obtained without taking the structure of the source article into account so that the last sentence of each excerpt can be incomplete. Moreover dates present in the excerpts were removed, in order to circumvent the bias of dates available within the document itself.

Table 1 summarizes general statistics about the corpora.[3] The training corpus provided by DEFT contains 3,596 newspaper portions. We divided this corpus in two parts: an actual training set (TRN) and a development set (DEV). The evaluation corpus (EVAL) was unavailable at the time of system development and contains 2,445 portions.

The corpora were automatically digitized with OCR. Figure 1 shows an example of digitized text in which erroneous words are underlined, while Figure 2 shows the original corresponding document.

Different kinds of errors can be identified, such as erroneous uppercasing, additional and/or missing letters, punctuation, or space, sequence of one or several erroneous letters... There are also archaic forms of words, such as "fragmens". We

[2] `http://gallica.bnf.fr/`

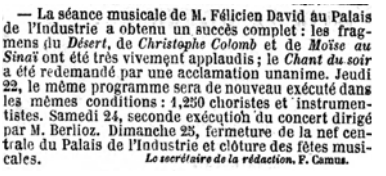
[3] The number of portions per year is 24 for each year except for 1815: 21 portions were proposed in the training set and 17 in the evaluation set.

**Table 1.** General description of training and test corpora

| | Training data | | | | Evaluation data | |
|---|---|---|---|---|---|---|
| | 300 words | | 500 words | | 300 words | 500 words |
| | TRN | DEV | TRN | DEV | EVAL | EVAL |
| # portions | 2396 | 1200 | 2396 | 1200 | 2445 | 2445 |
| # words | 718,800 | 360,000 | 1,198,000 | 600,000 | 733,500 | 1,222,500 |
| # different words | 73,921 | 48,195 | 107,617 | 67,012 | 78,662 | 110,749 |
| # different newspapers | 6 | 6 | 6 | 6 | 7 | 7 |
| Mean # portions per year | 16 | 8 | 16 | 8 | 14 | 14 |

*La séance musicale de M. Félicien David au Palais de l'Industrie a obtenu un succès complet les fragmens du Désert, de Christophe Colomb et de Moïse au Sinaï ont été très vivemçnt applaudis; le Chant du soir a été redemandé par une acclamation unanime. Jeudi 22, le même programme sera de nouveau exécuté dans les mêmes conditions: l,2S0 choristes et instrumentistes. Samedi 24, seconde exécution du concert dirigé par M. Berlioz. Dimanche 2S, fermeture de la nef centrale du Palais de l'Industrie et clôture des fêtes musicales. Lotecfètairedela rédaction, F. Carani.*

**Fig. 1.** Digitized text from a 1855 document



— La séance musicale de M. Félicien David au Palais de l'Industrie a obtenu un succès complet : les fragmens du *Désert*, de *Christophe Colomb* et de *Moïse au Sinaï* ont été très vivement applaudis ; le *Chant du soir* a été redemandé par une acclamation unanime. Jeudi 22, le même programme sera de nouveau exécuté dans les mêmes conditions : 1,250 choristes et instrumentistes. Samedi 24, seconde exécution du concert dirigé par M. Berlioz. Dimanche 25, fermeture de la nef centrale du Palais de l'Industrie et clôture des fêtes musicales.

*Le secrétaire de la rédaction*, F. Camus.

**Fig. 2.** Excerpt from a 1855 document

estimated the number of out of vocabulary (OOV) words using a contemporary spell checker: *hunspell*.[4] There are between 0 and 125 OOV words in 300-word portions and a mean of 22 OOV words per portion. We observed that there is no clear correlation between the publication year of an excerpt and the number of OOV words, i.e., the quality of the OCR document.

This kind of text is especially challenging for NLP tools, since traditional techniques such as part-of-speech tagging or named entity recognition are likely to have much lower performance on these texts.

### 3.2 Corpus Pre-processing

The corpus was preprocessed by the TreeTagger [17] for French, and words were replaced by their lemmas. The goal was to reduce the vocabulary, to improve the similarity between documents. For the portions of the TRN corpus for example, the vocabulary thus dropped from 74,000 to 52,000 different words.

### 3.3 Evaluation Score

The evaluation measures that we use for our final system are the percentages of correct decades and years given by our systems. Yet the aim is to be as close as

[4] Open source spell checker: `http://hunspell.sourceforge.net/`

possible to the reference year so we also use an evaluation metric which takes into account the distance between the predicted year and the reference year, which is the official DEFT 2011 evaluation score [5]. Given a text portion $a_i$ whose publication year in the reference is $d_r(a_i)$, a system gives an estimated publication date $d_p(a_i)$. The system then receives a score $S$ which depends on how close the predicted year is to the reference year. This similarity score is based on a gaussian function and is averaged on the N test portions. The precise formula is given by equation 1.

$$S = \frac{1}{N} \sum_{i=1}^{N} e^{-\frac{\pi}{10^2}(d_p(a_i) - d_r(a_i))^2} \tag{1}$$

This score is thus a variant of the fraction of correctly predicted years, where wrong predictions at a certain distance from the correct answer are given less points than correct answers, instead of no point as do more traditional measures. For example, the score is of 1.0 if the predicted year is correct, of 0.97 if off by one year, of 0.5 if off by 4.7 years, and falls to 0 if it is off by more by 15 years.

### 3.4 Description of the Methods

We used two types of methods. Chronological methods (see section 4) yield the periods of time which are most plausible for each portion, but without ranking the corresponding years. In the above example (Figure 1), several cues give indications on the publication date of the document: several persons are mentioned ("M. Félicien David" and "M. Berlioz" for example), which means that the publication date is (at least) posterior to their birthdates; moreover, the spelling of the word "fragmens" is an archaism, since it would now be written "fragments", which means that the text was written before the spelling reform modifying this word; finally, the exhibition hall "Palais de l'Industrie" was built in 1855 and destroyed in 1897, so the document date must be posterior to 1855, and is likely to be anterior to 1897 (as word statistics over time such as Google Books Ngrams can show). These are the kinds of information used by chronological methods to reduce the possible time span. These methods make use of external resources, and are thus not dependent on the corpora used.

Classification methods (see section 5) make use of the training corpora to calculate temporal similarities between each portion and a reference corpus.

## 4 Chronological Methods

### 4.1 Named Entities

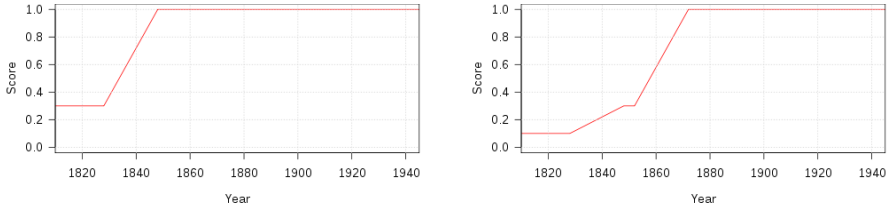
The presence of a person's name in a text portion is an interesting clue for determining its date, since the date of the document must be posterior to the birthyear of this person.

We used the following strategy: we automatically gathered the birthyears of persons born between 1781 and 1944 by using Wikipedia's "Naissance_en_$AAAA$"

categories.[5] About 99,000 person names were thus retrieved, out of which we selected 96,000 unambiguous ones (for example two "Albert Kahn" were found), since we have no simple way of knowing which particular instance is mentioned in the texts.

For each text portion, we extracted occurrences of person names using WMatch,[6] which allows for fast text annotation [4,16]. For the TRN corpus, 529 names were detected in 375 portions (out of 2,359 portions), out of which 16 (3%) were actually namesakes or false detections (for example, Wikipedia has an entry for the French novelist "Colette", whose name is also a common first name).

A score was then given to each candidate year for a given portion, according to the person mentions found in that portion. We considered that before the person birthyear $Y_b$, the probability of a year $y < Y_b$ being the correct answer is low (here 0.3), then for a year $y$ between the birthyear and 20 years after[7] ($Y_b \leq y \leq Y_b + 20$), the probability raises linearly reaching 1.0 (see Figure 3a).




(a) with Jules Verne (b) with Jules Verne and Antoni Gaudí

**Fig. 3.** Scoring function given person mentions

For a given text portion $p$, the score for each year is the product of the score for each person mention found in $p$. Figure 3b shows the score obtained in the presence of two person mentions, Jules Verne, born in 1828 and Antoni Gaudí, born in 1852.

### 4.2 Neologisms and Archaisms

Neologisms correspond to newly created words, while archaisms refer to words which cease being used at some time. Both neologisms and archaisms constitute interesting cues for identifying publication dates: given the approximate year of apparition of a word, one can assign a low probability for all preceding years and a high probability to following years (the reverse line of argument can be applied to archaisms). However, there is no pre-compiled list of words with their year

[5] Category:$YYYY$_birth.

[6] Rule-based automatic annotation tool, available upon request.

[7] Intuitively, a person that is less than 20 years old will not be cited in a newspaper and, in the absence of a more appropriate model, we considered that then s/he has a equal probability to be cited all over his/her life.

of appearance or disappearance. This type of information is sometimes included in dictionaries, but depends on the availability of these resources. We therefore developed a method to automatically extract neologisms and archaisms from Google Books unigrams for French [10].

**Automatic Acquisition of Neologisms and Archaisms.** Automatically determining the date of appearance and disappearance of a word is not a trivial task. In particular, metadata associated with Google Books are not always precise [14]. It is therefore not possible to use a simple criterion such as extracting the first year when the occurrence count of a word exceeds 1 to identify neologisms. We developed instead a method relying on the cumulatice frequency distribution, i.e., for each year, the number of occurrences of the word since the beginning of the considered time span divided by the total number of occurrences:

1. Get the word's count distribution for years ranging from 1700 to 2008;[8]
2. Smooth the distribution with a flat smoothing window[9] of size 3;
3. Get the word's cumulative frequency distribution and determine the appearance/disappearance date as the first year where the cumulative frequency exceeds a given threshold.

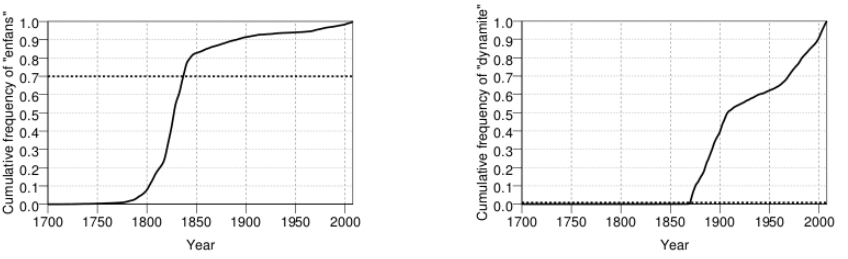
We defined the best cumulative frequency thresholds by using manually selected development sets consisting of 32 neologisms (e.g. "photographie" – *photography*, "télévision" – *television*) and 21 archaisms (old spellings which are no longer in use, see Section 4.3). This number of neologisms and archaisms was sufficient to find reliable thresholds. The obtained thresholds were 0.008 for neologisms and 0.7 for archaisms. Moreover, we only kept neologisms with a mean occurrence count of at least 10 and archaisms with a mean occurrence of at least 5 over the considered year range. Overall, we were able to extract 114,396 neologisms and 53,392 archaisms with appearance/disappearance year information.

Figure 4 displays two cumulative frequency curves: one for an archaism (the old spelling of the word "enfants", *children*), and the other for a neologism ("dynamite", invented in 1867). The thresholds correspond to the horizontal dotted lines. The curves have very different profiles: archaisms are characterised by a logistic curve, which reaches a plateau well before the end of the considered year range. On the other hand, neologisms correspond to an increasing curve.

We calculated the error rate on the DEV corpus: for 90% of the archaisms found in the corpus, the date of the portion is anterior to the disappearance date, and for 97% of them, it is anterior to the disappearance date plus 20 years. For the neologisms, the date of the portion is posterior to the appearance date for 97% of them, and to the appearance date minus 20 years for 99.8% of them. This 20-years *"shift"* (20 years giving the most accurate and precise results on the training corpus) is taken into account in the scoring formula.

[8] The first available year in Google Books ngrams is actually 1536. However, given the year-range of our task, we considered that 1700 was an adequate lower threshold.

[9] As defined in `http://www.scipy.org/Cookbook/SignalSmooth`




(a) Archaism (b) Neologism

**Fig. 4.** Cumulative frequency distributions

**Scoring with Neologisms and Archaisms.** The automatically extracted lists of neologisms and archaisms are used to assign a score for each year, given a text portion. For neologisms, years following the appearance date are given a high score, while preceding years are assigned a lower score. The following formula is used for neologisms. $p$ corresponds to text portion, $w$ is a word, $y$ a year in the considered year range 1801-1944 and year($w$) is the date of appearance extracted for a neologism.

$$score_{neo}(p,y) = \frac{\sum_{w \in p} score\text{-}neo(w,y)}{|p|} \text{ where: } score\text{-}neo(w,y) =$$

$$\begin{cases} 1.0 & \text{if } w \notin \text{neologisms} \\ 1.0 & \text{if } w \in \text{neologisms and } y \geq year(w) \\ 0.2 & \text{if } w \in \text{neologisms and } (year(w) - y) > 20 \\ 0.2 + 0.04 \cdot (20 + y - year(w)) & \text{otherwise} \end{cases}$$

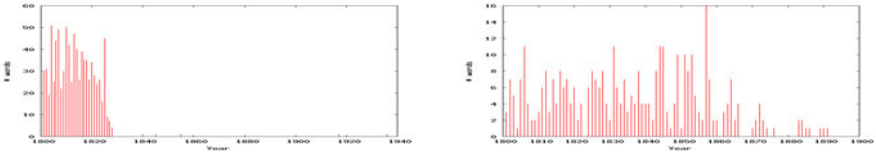
An equivalent formula is used for archaisms, by considering that years following the disappearance of a word have a low score.

### 4.3 French Spelling Reforms

During the 1801-1944 period, French spelling underwent two major reforms: one in 1835 and another in 1878. The main change induced by the first reform is that conjugated verbs ending with "oi" changed to "ai": e.g. the inflected form "avois" of the verb "avoir" (*to have*), was changed into "avais". The second reform mostly concerned names ending with "ant" or "ent", whose plural changed to "ants"/"ents" instead of "ans"/"ens" (for example "enfans" was changed into "enfants"–*children*).

Figure 5 displays the distribution of each type of words ("oi" and "a/ents") in the training corpus for each year. The first type of words is present mostly before 1828, and the second type only before 1891, which roughly correspond to the reform dates.

**Scoring with Spelling Reforms.** Following Albert et al. [1], we use this information as a clue to determine the date of a text. We assign a score for each



(a) Words ending with "oi" (b) Words ending with "a/ents"

**Fig. 5.** Distributions of pre-reforms words in the TRN corpus

year to each text portion. In order to determine old spellings in use before the reforms, we use the following method:

- Get unknown words with *hunspell* (with the French DELA as a dictionary [2]);
- If the word ends with "ois/oit/oient", replace "o" with "a";
  If the new word is in the dictionary, increment the counter $n_{28}$, which corresponds to the number of old word spellings in use before the first reform;
- Else, if the word ends with "ans/ens", insert "t" before "s";
  If the new word is in the dictionary, increment the counter $n_{91}$, which corresponds to the number of old word spellings in use before the second reform.

Then, a function was used to determine a score for each year $y$ and a portion $p$ based on the counters $n_{28}$ and $n_{91}$, according to the following formulas (where $r$ in $f_r$ can be either 28 or 91):

$score_{spell}(p, y) = score_{28}(p, y) \cdot score_{91}(p, y)$ with:

$$score_r(p, y) = \begin{cases} f_r(y) & \text{if } y > r \\ 1 & \text{if } y \leq r \end{cases}, \quad f_{28}(y) = \begin{cases} 1 & \text{if } n_{28} = 0 \\ 0.15 & \text{if } n_{28} = 1 \\ 0 & \text{if } n_{28} > 1 \end{cases} \text{ and } \quad f_{91}(y) = \begin{cases} 1 & \text{if } n_{91} = 0 \\ 0 & \text{if } n_{91} > 0 \end{cases}$$

For example, if $n_{28} = 1$ and $n_{91} = 1$ for a text portion, the score for years before 1828 is 1.0, for years between 1828 and 1891, the score is 0.15, which corresponds to the error rate for using this criterion on our training corpus, and for years after 1891, the score is 0 since the presence of an old spelling in use before the second reform is a very strong indication that the text was written before 1891.

### 4.4 Intermediate Conclusion

As we have shown in the previous section, chronological methods yield very accurate indications for a text's time span (with a maximum error rate of 3%). However, they only discriminate between large time periods, and are not precise enough for identifying the publication date (e.g. if a portion contains a person's name whose birthyear is 1852, we can only say the portion has not been published before 1852). Thus, we also used corpus-based classification methods: a cosine

similarity relying on a feature vector representation of text portions and using the standard $tf \cdot idf$ feature weighting; and a machine learning approach based on SVMs. These approaches are described in next sections.

## 5 Classification Methods

Temporal similarity methods calculate similarities between each portion and a reference corpus.

### 5.1 Cosine Similarity-Based Classification

**Using the Training Corpus.** The training corpus provides examples of texts for each year in the 1801-1944 year range. These texts can be used as a reference to obtain word statistics over time. We grouped all portions for the same year in the TRN corpus and used these portion groups as references for the corresponding years. For classification, the similarity is computed between a group of portions in the same year and the portion to be classified. Each group and each portion were converted into feature vectors using the $tf \cdot idf$ score as feature weighting. Given an n-gram $i$ and a portion (or group of portions) $j$:

$$tf \cdot idf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \cdot \log \frac{|Y|}{|\{y_j : w_i \in y_j + smoothing\}|}$$

where $n_{i,j}$ is the number of occurrences of n-gram $w_i$ in portion (or group) $j$, $|Y|$ is the number of years in the training corpus, $y_j$ is the group of text portions for year $j$; $smoothing = 0.5$ is applied to take into account words in the test corpus which were not found in the training corpus.

For a text portion in the test corpus, we computed the similarities between the portion and each group representing a year with a standard cosine similarity. Experiments were made for word $n$-grams with $n$ ranging from 1 to 5; yet, for n>2, the small size of the training corpus leads to sparse data. For word $n$-grams, we used the lemmatized version of the corpora since it gave better results in preliminar experiments.

As the corpus is composed of OCRized documents, there are many errors in the texts, which poses many problems for tf.idf scoring: the tfs and dfs are smaller than what would be expected for "real" words since errors impede the identification of some occurrences, and some erroneous words have higher idfs than would be expected. In order to cope with this difficulty, we also computed the similarity using character $n$-grams (following [12] for information retrieval on an OCRized corpus). Thus, for example for the text "sympathie1" which contains a "real" word "sympathie" and an OCR error "1", character $n$-grams (for n<9) will match all $n$-grams of the word "sympathie", despite the OCR error. Then, portions were indexed as before, and a cosine similarity was also applied to match each portion with the best corresponding year.

**Using Google Books Ngrams.** The training corpus is rather small, and we therefore also experimented with using Google Books Ngrams as training data. Due to the huge amount of data in Google Books Ngrams, we only used the $n$-grams with alphanumeric content and with more than 10 occurrences in a given year. The resulting data was used instead of our training corpora. The tf.idf formula is slightly modified for the training corpus, since $n_{i,j}$ is the number of occurrences of $n$-gram $w_i$ for year $j$ and $y_j$ is the Google Ngram data for year $j$.

### 5.2 Support Vector Machines (SVM)

SVMs are well-known machine learning algorithms belonging to the class of maximal margin linear classifiers [18]. For our experiments with SVM we used `svm-light`[10] [7]. Two kernel functions have been tested for our task: `polynomial kernel` and `radial basis function`, both available in the `svm-light` package. Given the small amount of data available for each year (25 portions for each year, except for 1815 which has 21 portions), the one-VS-all training approach was used: a model was created for each year against all other years. The SVM system consists of 144 binary models, one corresponding to each year, from 1801 to 1944. In each model, positive instances are those extracted from portions belonging to the target year to be detected, negative instances are all the others. Each model is able to distinguish portions belonging to the corresponding year. At classification time, each portion is evaluated with all 144 models and the one providing the highest score is chosen as the correct answer.

**SVM Settings and Tuning.** SVM parameters as well as feature sets were tuned on the TRN and DEV sets. Neither all parameters, nor all features types were optimized. A full optimization of all parameters and features requires a huge number of experiments. Instead, based also on our experience, in some cases we used default or a-priori parameters. The SVM parameter $C$ for soft margin (see [18]) was set to 1. In most of the tasks the best value is between 1 and 10, 1 gives always fair results. The *cost-factor* parameter, affecting the weighting of errors made on positive and negative instances, was set to the ratio between the number of negative and positive instances, as suggested in [11]. Concerning kernel functions, the `polynomial kernel` was more effective than the `radial basis function` on the DEV set and it was kept for following system tuning. Default values for polynomial kernel parameters were used (1 for $c$ and 3 for polynomial degree $d$).

Concerning the feature set, we tried several sets for preliminary studies, and for further experiments we kept only the most promising in terms of performance on DEV. We first experimented with some configurations typical of text categorization tasks. For example we removed stop-words and we replaced words by their lemmas (in inflectional languages like French, they provide roughly the same information as stems). Surprisingly this led to a degradation of performances. In contrast, using both words and lemmas and keeping stop-words,

[10] Available at `http://svmlight.joachims.org/`

gave better results than those obtained using only words. This configuration was chosen as baseline SVM system. Further experiments were performed to tune the size of word $n$-grams to be used in feature vectors. We tried to use $n$-grams of size from 1 up to 4. 2-grams gave best results.

Using this configuration we integrated the information provided by systems described in section 4: birth dates of persons, neologisms and archaisms, French spelling reforms. In particular each of these systems provides information that could be encoded in SVM feature vectors as *feature:year*, where *feature* is a person name in case of birth dates, a neologism or archaism word or a word that has been reformed in one of the two French spelling reforms. Given the sparsity of feature vectors representation, feature values in the baseline system are always much smaller than any of the *year* provided by any of the chronological methods. This has been a problem for learning the SVM models. The problem still holds when shifting year values from the range 1801..1944 to the range 1..144. Indeed we experienced training problems or performance degradation when using such a representation. In order to overcome this problem we split the information provided by chronological methods in two parts, corresponding to two sets of binary features (the value is 0 if the feature is absent, 1 if present): one for the information alone, e.g. *NEOLOGISM_ <WORD>* or *REFORMED_ <WORD>* for neologisms or reformed words,[11] respectively; another for the year the information appears in, e.g. *NEOLOGISM-YEAR_ <YEAR>* or *REFORMED-YEAR_ <YEAR>*. This representation always led to performance improvements.

Since in preliminar studies experiments on 500-word portions reflected the behavior of 300-word portions, we did not carry out all experiments also on 500-word portions. Instead we applied directly the best configuration found for 300-word portions.

## 6 Scoring Combination

Given the differences in characteristics of individual systems described in previous sections, we made a combination of the score provided by each individual system with the aim of improving the final result. The methods do not have the same overall performance nevertheless they all provide useful information: for instance, archaisms indicate an upper limit for the publication date. For the combination of scores, we experimented with two different strategies: simple multiplication and linear regression of scores provided by individual systems.

**Multiplication of Scores.** This combination consists in multiplying the scores provided by the different methods, for each portion and for each year:

$$score_{multiplication}(p, y) = \prod_k score_k(p, y)$$

where $score_k(p, y)$ is the score of the system $k$ labelling portion $p$ as being published in year $y$.

[11] <WORD> is a place holder for any word belonging to the specified category.

**Linear Regression on Scores.** In this case, scores from different systems are not multiplied but summed according to the following formula:

$$score_{regression}(p, y) = \sum_k \alpha_k \cdot score_k(p, y) + \varepsilon$$

with $\alpha_k$ the coefficient for the system $k$, $score_k(p, y)$ the score given by the system $k$ to the portion $p$ for the year $y$ and $\varepsilon$ the error term.

Coefficients were fitted on the training corpus using the R function `lm()`. The linear regression process finds the best model (ie. $\alpha$ values) to predict a numerical value from clues (system scores in our case). In our case, the numerical value to be predicted depends on the distance *dist* between a year and the true year of publication of the portion : the value is $1.0 - dist/143$.

In the development phase, we fitted the $\alpha$ and $\varepsilon$ values on the TRN corpus and tested the combination on the DEV corpus. As the cosine and SVM systems need to be trained, we did not include the score of those systems in our regression model. We thus computed a *regression score* based on scores from neologism, archaism, birth dates of person, and spelling reforms information. The scores of the cosine and SVM systems were multiplied by this regression score. For the test phase, we fitted the values on the entire training data set.

## 7 Results

We evaluate our approach using the measures described in section 3.3. We first present the results of the cosine and SVM approaches and then the results of the two scoring combination methods described in section 6. The systems used for the evaluation data have been trained on the entire training data (TRN + DEV).

### 7.1 Results for Classification Methods

**Cosine Similarity.** The results of the cosine similarity system are presented in table 2 (only the best scoring settings are given). With the training corpus, characters 5-grams have the best results on both portion sizes, which was expected since the documents are quite noisy. Word unigrams are better on 300-word portions than bigrams. Yet bigrams perform better on 500-word portions, which tends to show that they benefit from an increased amount of data.

**Table 2.** Results obtained for the cosine based methods

| | **Training corpus** | | | | **Google Ngrams** | | | |
|---|---|---|---|---|---|---|---|---|
| | **DEV** | | **EVAL** | | **DEV** | | **EVAL** | |
| | **300 w.** | **500 w.** | **300 w.** | **500 w.** | **300 w.** | **500 w.** | **300 w.** | **500 w.** |
| **word 1-grams** | 0.260 | 0.299 | 0.267 | 0.321 | 0.210 | 0.221 | 0.200 | 0.216 |
| **word 2-grams** | 0.209 | 0.319 | 0.263 | 0.327 | 0.238 | 0.295 | 0.241 | 0.264 |
| **char 5-grams** | **0.287** | **0.327** | **0.311** | **0.363** | - | - | - | - |

For the cosine method based on Google Ngrams, the corpus used was not lemmatized, since Google Ngrams contain inflected words. The best results were obtained with bigrams. Results are lower than those using only the training corpus which was not expected because Google Ngrams is a much larger data set. This could be due to the different nature of documents: our corpus is composed only of newspaper excerpts. Moreover the publication dates in Google Books are not completely reliable [14].

**SVM System.** Results obtained with the system based on SVM are reported in tables 3 and 4. As can be seen from table 3, incrementally adding features encoding the information provided by chronological methods leads to consistent performance improvements. In table 4 we detail all the results obtained with the best system on 300 and 500-word portions.

**Table 3.** Additive results of the SVM system with different features on the DEV corpus for 300 words

| | |
|---|---|
| **Baseline** (word 2-grams + lemmas) | 0.228 |
| **+neologisms** | 0.234 |
| **+spelling reforms** | 0.242 |
| **+birth dates** | 0.243 |

**Table 4.** Results of SVM system

| DEV | | EVAL | |
|---|---|---|---|
| **300 words** | **500 words** | **300 words** | **500 words** |
| 0.243 | 0.293 | 0.272 | 0.330 |

| | DEV | | EVAL | |
|---|---|---|---|---|
| | **300 w.** | **500 w.** | **300 w.** | **500 w.** |
| **mult.** | 0.343 | 0.401 | 0.378 | 0.452 |
| **regress.** | 0.356 | 0.390 | 0.374 | 0.428 |

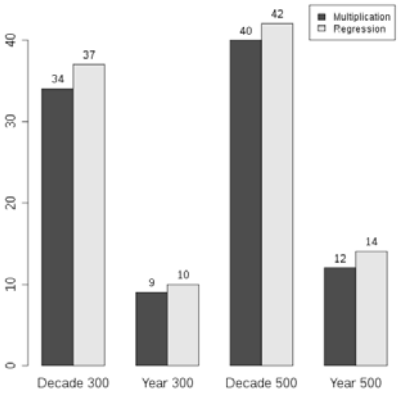



**Fig. 6.** Scores and correct decades/years obtained with fusion

**Scoring Fusion.** Figure 6 displays the results obtained on the training and evaluation data sets for the various system combinations. Scoring fusions consistently improve the scores of individual systems. Results on 500-word portions are much higher than results on 300-word portions. For the evaluation data, fusion by multiplication performs better than fusion using linear regression.

Figure 6 shows results in terms of correct decades and years at the first rank. 35% of first rank decades are the correct ones for 300-word portions and 40% for 500-word portions. For years, the fusion using linear regression detects the correct year for respectively 10% and 14% of the 300 and 500-word text portions. Those results are much higher than the random selection of a decade or a year in the time span (7% for decades and 0.7% for years). For decades, using the DEFT 2010 evaluation metric, our results are also higher than results obtained by the best participants to the DEFT 2010 challenge [6].

## 8 Conclusions and Future Work

In this article, we present a system for automatically dating historical documents. It is based on several methods, both supervised and unsupervised, and takes advantage of different external resources, such as Google Ngrams or knowledge about spelling reforms. We obtain 14% of correct years and 42% of correct decades in our best-performing setting.

The results show that this is a challenging task for several reasons: the documents may not contain many intra-textual hints about their publication dates, digitized historical documents can be of a low quality, the vocabulary is different from the vocabulary currently in use, and external resources are not always completely reliable.

These experiments made it possible to observe the quality of digitized documents, and to adapt the NLP techniques we used to this specific condition, for example by considering characters $n$-grams instead of word $n$-grams. In order to improve the quality of documents, we plan to use OCR correction. We would also like to investigate the application of named entity recognition, including event detection. Finally, we plan to work on different corpora in order to test the robustness of our methods, and to perform experiments with whole documents without date anonymisation instead of text portions.

## References

1. Albert, P., Badin, F., Delorme, M., Devos, N., Papazoglou, S., Simard, J.: Décennie d'un article de journal par analyse statistique et lexicale. In: DEFT 2010, TALN (2010)
2. Blandine, C., Silberzstein, M.: Dictionnaires électroniques du français. Langue française 87 (1990)
3. De Jong, F., Rode, H., Hiemstra, D.: Temporal language models for the disclosure of historical text. In: Humanities, Computers and Cultural Heritage, p. 161 (2005)
4. Galibert, O.: Approches et méthodologies pour la réponse automatique à des questions adaptées à un cadre interactif en domaine ouvert. Ph.D. thesis, Université Paris-Sud 11, Orsay, France (2009)
5. Grouin, C., Forest, D., Paroubek, P., Zweigenbaum, P.: Présentation et résultats du défi fouille de texte DEFT2011. In: Actes TALN (2011)
6. Grouin, C., Forest, D., Sylva, L.D., Paroubek, P., Zweigenbaum, P.: Présentation et résultats du défi fouille de texte DEFT 2010: Oú et quand un article de presse a-t-il été écrit? In: Actes TALN (2010)

7. Joachims, T.: Making large-scale SVM learning practical. In: Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1999)
8. Kanhabua, N., Nørvåg, K.: Improving temporal language models for determining time of non-timestamped documents. In: Research and Advanced Technology for Digital Libraries, pp. 358–370 (2008)
9. Kanhabua, N., Nørvåg, K.: Using temporal language models for document dating. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5782, pp. 738–741. Springer, Heidelberg (2009)
10. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., The Google Books Team, Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A., Aiden, E.L.: Quantitative Analysis of Culture Using Millions of Digitized Books. Science 331(6014), 176–182 (2011)
11. Morik, K., Brockhausen, P., Joachims, T.: Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In: Proceedings of ICML 1999, pp. 268–277. Morgan Kaufmann Publishers Inc., San Francisco (1999)
12. Naji, N., Savoy, J., Dolamic, L.: Recherche d'information dans un corpus bruité (OCR). In: CORIA (2011)
13. Nørvåg, K.: Supporting temporal text-containment queries in temporal document databases. Data & Knowledge Engineering 49(1), 105–125 (2004)
14. Nunberg, G.: Google's Book Search: A Disaster for Scholars. The Chronicle of Higher Education (August 2009) (Online, accessed April 13, 2011)
15. Oger, S., Rouvier, M., Camelin, N., Kessler, R., Lefèvre, F., Torres-Moreno, J.: Système du LIA pour la campagne DEFT 2010: datation et localisation d'articles de presse francophones. In: DEFT 2010, TALN (2010)
16. Rosset, S., Galibert, O., Bernard, G., Bilinski, E., Adda, G.: The LIMSI participation to the QAst track. In: Working Notes of CLEF 2008 Workshop, Aarhus, Danemark (2008)
17. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: International Conference on New Methods in Language Processing, pp. 44–49 (1994)
18. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Chichester (1998)