An SVM-based approach for Automatic Document Dating

Matteo Brivio
University of Tübingen
Department of Linguistics
matteo.brivio@student.uni-tuebingen.de

Abstract

English. This paper describes our contribution to the Evalita 2020 shared task DaDoEval – Dating Document Evaluation. The solution we present is based on a linear multi-class Support Vector Machine classifier trained on a combination of character and word n-grams, as well as number of word tokens per document. Despite its simplicity, the system ranked first both in the coarse-grained classification task on same-genre data and in the one on cross-genre data, achieving a macro-average F1 score of 0.934 and 0.413, respectively. The system implementation is available at https://github.com/matteobrv/DaDoEval.

Italiano. *Questo articolo descrive un modello per la datazione automatica di documenti presentato per la partecipazione al task DaDoEval – Dating Document Evaluation – proposto nell'ambito di Evalita 2020. Il sistema consiste in un classificatore lineare multiclasse implementato con una macchina a vettori di supporto e fa uso di tre insiemi di feature: n-grammi di caratteri, n-grammi di parole e lunghezza di ogni documento espressa in numero di parole. Per quanto elementare, il sistema ottiene il primo posto sia nella prima sub-task, con un punteggio F1 pari a 0.934, sia nella seconda, con un punteggio di 0.413. Il codice è liberamente consultabile* https://github.com/matteobrv/DaDoEval.

1 Introduction

Temporal information, such as the publication date of a document, is of major relevance in a number of domains, like historical linguistics and digital humanities (Niculae et al., 2014). This is arguably even more true for a wide range of information retrieval tasks, such as document exploration, similarity search, summarisation and clustering, where the temporal dimension plays a major role in improving search results (Alonso et al., 2007; Alonso et al., 2011). Such information, however, is not always readily available and must therefore be inferred, relying either on qualitative or quantitative methods, if not both (Ciula, 2017). Nonetheless, despite their significance, methods for temporal text classification and automatic document dating are still rather unexplored compared to other text classification tasks (Niculae et al., 2014). This, however, is most likely bound to change as the increasing availability of large-scale, time-annotated digital resources, such as Google n-grams[1], is promoting research in this direction. Two recent examples of this new trend, in line with the present task, are the Diachronic Text Evaluation shared task organised by Popescu et al. (2015) at SemEval 2015 and the The RetroC challenge described by Graliński et al. (2017).

In this work we propose a simple, yet effective, approach for automatic document dating based on a linear multi-class Support Vector Machine classifier, trained on a combination of character and word n-grams, as well as document length in word tokens.

The solution is evaluated in the context of the DaDoEval – Dating Document Evaluation – shared task at Evalita 2020 (Menini et al., 2020). The task is based on the Alcide De Gasperi's corpus of public documents (Tonelli et al., 2019) and is organised into six sub-tasks: (I) coarse-grained classification on same-genre data, (II) coarse-grained classification on cross-genre data, (III) fine-grained classification on same-genre data, (IV) fine-grained classification

[1] http://books.google.com/ngrams

	1901-1918	*1919-1926*	*1927-1942*	*1943-1947*	*1948-1954*
SAMPLES PER CLASS	572	342	150	514	632
AVG. SAMPLE LENGTH	867	1033	3044	633	1209

Table 1: Data set overview, showing the number of document samples per class and the average number of word tokens per sample, rounded up to the nearest integer.

on cross-genre data, (V) year-based classification on same-genre data, (VI) year-based classification on cross-genre data.

The proposed solution tackles the first two sub-tasks, coarse-grained classification on same-genre and cross-genre data. Both sub-tasks require to correctly assign document samples to one of the main five time periods identified in De Gasperi's political life, spanning a range of over fifty years from 1901 to 1954.

The paper is structured as follows: in section 2 we provide a brief overview of the training data set, in section 3 we go over the system setup and describe the feature space, section 4 is dedicated to results analysis and discussion, in section 5 we consider possible improvements while section 6 is reserved for final remarks.

2 Data

The training data set released for the shared task is outlined in Table 1. It includes 2,210 document samples extracted from the Alcide De Gasperi's corpus of public documents, a multi-genre collection of 2,762 texts written or transcribed between 1901 and 1954 (Tonelli et al., 2019).

With respect to the coarse-grained classification sub-tasks, the given samples are organised into five classes corresponding to the main time periods historians identified in De Gasperi's political life: *Habsburg years* 1901-1918, *Beginning of political activity* 1919-1926, *Internal exile* 1927-1942, *From fascism to the Italian Republic* 1943-1947, *Building the Italian Republic* 1948-1954.

A preliminary analysis of the data set reveals an imbalanced class distribution, with a significantly lower number of samples in the third class, corresponding to the 1927-1942 interval. This, however, is partially mitigated by the markedly higher average number of word tokens per sample observed in this class compared to the other ones.

3 System Description

The proposed solution is based on a Support Vector Machine (SVM) classifier implemented using the Scikit-learn library (Pedregosa et al., 2011).

To account for the rather imbalanced data set, the SVM is tuned in such a way that classes are assigned weights inversely proportional to their frequency in the input data.

Following the assumption that most text categorisation problems are linearly separable (Joachims, 1998) the model uses a linear kernel implemented in terms of `libsvm` (Chang and Lin, 2011) while relying on a `one-versus-one` decision strategy to handle both sub-tasks as multi-class, single label, classification problems.

3.1 Feature space

The system relies solely on the data provided by the task organisers and no preprocessing is applied, as measures such as case normalisation and punctuation removal do not seem to improve the classification result on the development set, but rather to worsen it.

Each document in the data set is represented using three sets of features: document length in terms of word tokens as well as character and word n-grams. In this latter respect, we explore the idea that SVMs trained on combinations of character and word n-grams are particularly effective in tackling text classification tasks (Çöltekin and Rama, 2017; Çöltekin and Rama, 2018).

Character n-grams are extracted for $n \in \{3, 4, 5\}$ and span across word boundaries, thus capturing punctuation and space characters occurring at the beginning and at the end of each word token. Word n-grams, on the other hand, are extracted for $n \in \{1, 2\}$. Both feature sets are weighted using term-frequency, inverse-document frequency (`TF-IDF`) to scale down the impact of the most frequent n-grams.

The number of word tokens per document is computed in a naive way, splitting each sample at

every white space. Similarly to n-gram features, tokens count are scaled down to a 0-1 range in an attempt to avoid numerical problems and prevent features in higher numeric ranges from dominating those in smaller ones (Hsu et al., 2003).

3.2 Optimisation and Tuning

The system hyper-parameters are optimised to obtain the best F1 score on the development set which consists of 20% of the original training set.

A subset of the hyper-parameters is tuned empirically through several experiments or on the basis of existing literature. This is the case for kernel type, decision strategy, class balancing, tolerance for stopping criterion (`tol`) and n-grams size.

The remaining hyper-parameters considered during optimisation are the regularisation parameter (`C`) together with the maximum and minimum document frequency (`max_df`, `min_df`), which in the present approach are used to set an acceptance threshold for high and low frequency n-grams.

COMPONENT	PARAMETER	VALUE
TfidfVectorizer	analyzer	word
	max_df	0.9
	min_df	0.004
	ngram range	(1, 2)
	lowercase	False
TfidfVectorizer	analyzer	char
	max_df	0.3
	min_df	0.001
	ngram range	(3, 5)
	lowercase	False
SVM	kernel	linear
	decision function	ovo
	tol	1e-12
	C	0.881
	class weight	balanced

Table 2: Final hyper-parameters setup for each system component.

These hyper-parameters are tuned through the `BayesSearchCV` algorithm implemented in the scikit-optimize library (Head et al., 2020), using a 5-fold-shuffled cross validation. `BayesSearchCV` relies on Bayesian Optimisation and explores the hyper-parameters search space exploiting the information available from previous evaluations. This is in contrast to other approaches, such as grid and random search, which move across the search space either in an exhaustive or completely random manner.

Table 2 summarises the best hyper-parameters setup obtained from the tuning process.

4 Results

In this section we present the results for the two sub-tasks the system participated to. Results are summarised in Table 3 and reported in terms of macro-average F1 score.

SUB-TASK	TEAM	RUN	MACRO F1
same-genre	matteo-brv	1	0.934
		2	0.934
	team 1	1	0.858
		2	0.855
	baseline	-	*0.827*
cross-genre	matteo-brv	1	0.413
		2	0.413
	team 1	1	0.392
	baseline	-	*0.368*
	team 1	2	0.366

Table 3: Final rankings for sub-task 1 and 2 in terms of macro-average F1 scores.

The system ranked first both in the same-genre and in the cross-genre coarse-grained classification task, obtaining a macro-average F1 score of 0.934 and 0.413, respectively.

4.1 Classification on same-genre data

The runs submitted for the first sub-task are based on test samples of the same genre as the ones in the training set. The system scored well above the baseline, which was computed with a Logistic Regression model trained on `TF-IDF`-weighted word unigrams, without performing any preprocessing.

Overall, the results registered on the test set are in line with those observed during training. This is confirmed by the data summarised in Table 4 and by the confusion matrix in Figure 1.

The confusion matrix depicts a run on the development set which achieved a macro-average F1 score of 0.95, while Table 4 reports the per-class results of the best test run submitted for the sub-task. In both cases 1919-1926, 1943-1947 and 1948-1954 are the classes showing the highest number of misclassifications and, incidentally, are

also the ones corresponding to the shortest time periods.

CLASS	PRECISION	RECALL	F1
1901-1918	0.914	0.986	0.948
1919-1926	0.96	0.872	0.913
1927-1942	0.973	0.973	0.973
1943-1947	0.898	0.898	0.898
1948-1954	0.939	0.933	0.936

Table 4: Per-class results of the best test run for sub-task 1.

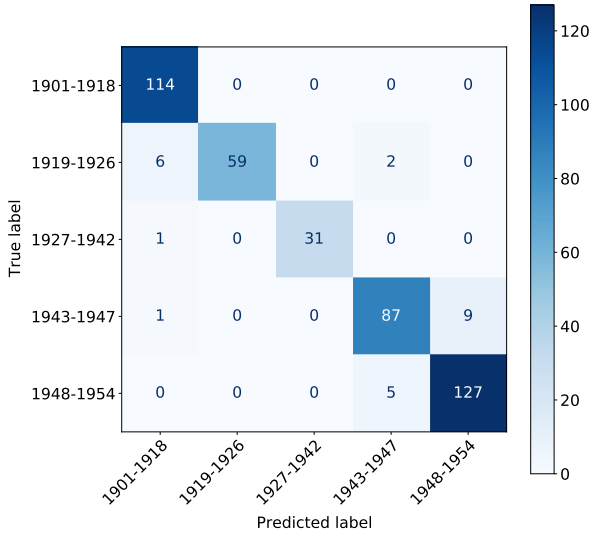

Figure 1: Confusion matrix for a development set run with a macro-average F1 score of 0.95.

4.2 Classification on cross-genre data

The runs submitted for the second sub-task are based on samples coming from a cross-genre, out-of-domain test data set. These samples are a subset of the documents collected for the Epistolario project (Tonelli et al., 2020), an ongoing effort to create a digital archive of Alcide De Gasperi's private and public correspondence.

CLASS	PRECISION	RECALL	F1
1901-1918	0.583	0.7	0.636
1919-1926	1.0	0.15	0.261
1927-1942	0.0	0.0	0.0
1943-1947	0.6	0.75	0.667
1948-1954	0.354	0.85	0.5

Table 5: Per-class results of the best test run for sub-task 2.

As expected, despite scoring above the baseline, cross-genre results are significantly lower than those obtained in the same-genre task. Per-class results summarised in Table 5 show how promising system performances registered in the same-genre task do not transfer to the cross-genre one, suggesting a poor ability of the model to generalise. Particularly interesting and worth investigating are the results registered for the third class, corresponding to the 1927-1942 interval. With respect to this class precision and recall values are equal to 0, indicating that model did not recognise any sample as belonging to this time period.

5 Possible improvements

Results for the same-genre task are quite encouraging and in line with those obtained on the development set, where the F1 score ranges between 0.92 and 0.96. However, with the current data and setup, there might not be much room for further improvement. Nonetheless, additional features like richness measures and linguistically motivated features (e.g. POS tags) are explored in other contributions (Štajner and Zampieri, 2013; Zampieri et al., 2016) and could help achieve more stable results.

On the other hand, results for the second sub-task suggest a lack of generalisation on cross-genre, out-of-domain data. In this respect, even though SVM-based systems for text classification should be able to perform well and take advantage of high dimensional feature spaces (Joachims, 1998), it might still be worthwhile experimenting with some feature selection methods. Another angle worth considering is that the system might be too sensitive to the shallow n-gram features used to represent the training data. In this case, including deeper text features, such as those encoding syntactic information, might help the system to abstract away from the lexical level. A first step in this direction is attempted by Szymanski and Lynch (2015) who employ Google Syntactic N-grams in an SVM-based system that participated to the Diachronic Text Evaluation shared task (Popescu et al., 2015) at SemEval 2015.

6 Conclusions

In this paper we describe a simple, yet effective, approach for automatic document dating implemented for the DaDoEval shared task at Evalita 2020. The system is based on a linear Support

Vector Machine and is trained on a small set of stylistic and lexical features, resulting in a fast and efficient classification model.

In particular, the approach achieves top scores in both coarse-grained classification sub-tasks, thus confirming that SVM-based systems trained on character and word n-grams are indeed well suited to tackle text classification problems.

Nonetheless, results observed in the second task suggest that the model does not generalise well on cross-genre data, leaving room for further improvements.

Acknowledgments

We thank Dr. Çağrı Çöltekin for his patient encouragement and valuable suggestions throughout this project.

References

Arianna Ciula. 2017. Digital palaeography: What is digital about it? *Digital Scholarship in the Humanities*, 32(2):ii89–ii105.

Çağrı Çöltekin, Taraka Rama. 2018. Tübingen-oslo at SemEval-2018 task 2: SVMs perform better than RNNs in emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, 34-38.

Çağrı Çöltekin, Taraka Rama. 2017. Tübingen system in VarDial 2017 shared task: experiments with language identification and cross-lingual parsing. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 146-155.

Chih-chung Chang, Chih-jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Chih-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin. 2003. A practical guide to support vector classification. *Technical report, Department of Computer Science, National Taiwan University*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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*, 29–34.

Marcos Zampieri, Shervin Malmasi and Mark Dras. 2016. Modeling Language Change in Historical Corpora: The Case of Portuguese. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, 4098–4104.

Octavian Popescu and Carlo Strapparava. 2015. Semeval 2015, task 7: Diachronic text evaluation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 870–878.

Omar Alonso, Strötgen Jannik, Baeza Y. Ricardo and Gertz Michael. 2011. Temporal Information Retrieval: Challenges and Opportunities. In *Proceedings of the 1st International Temporal Web Analytics Workshop*, 11:1–8.

Omar Alonso, Gertz Michael and Baeza Y. Ricardo. 2007. On the value of temporal information in information retrieval. *SIGIR Forum*, 41:35–41.

Sanja Štajner and Marcos Zampieri. 2013. Stylistic Changes for Temporal Text Classification. In *Proceedings of the 16th International Conference on Text, Speech and Dialogue (TSD), Lecture Notes in Artificial Intelligence - LNAI 8082, Springer*, 519-526.

Sara Tonelli, Rachele Sprugnoli and Giovanni Moretti. 2019. Prendo la Parola in Questo Consesso Mondiale: A Multi-Genre 20th Century Corpus in the Political Domain. In *Proceedings of CLIC-it 2019*.

Sara Tonelli, Rachele Sprugnoli, Giovanni Moretti, Stefano Malfatti and Marco Odorizzi. 2020. Epistolario De Gasperi: National Edition of De Gasperi's Letters in Digital Format. In *Proceedings of AIUCD*.

Stefano Menini, Giovanni Moretti, Rachele Sprugnoli and Sara Tonelli. 2020. Overview of the EVALITA 2020 task on dating documents (DaDoEval). In *Proceedings of EVALITA 2020*.

Terrence Szymanski and Gerard Lynch. 2015. UCD: Diachronic Text Classification with Character, Word, and Syntactic N-grams. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 879–883.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, 1398:137–142.

Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe and Iaroslav Shcherbatyi. 2020. scikit-optimize/scikit-optimize (Version v0.8.1). Zenodo *http://doi.org/10.5281/zenodo.4014775*.

Vlad Niculae, Marcos Zampieri, Liviu Dinu and Alina M. Ciobanu. 2014. Temporal Text Ranking and Automatic Dating of Texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2:17–21.