rmassidda @ DaDoEval: Document Dating Using Sentence Embeddings at EVALITA 2020

Riccardo Massidda
Università di Pisa
r.massidda@studenti.unipi.it

Abstract

This report describes an approach to solve the DaDoEval document dating subtasks for the EVALITA 2020 competition. The dating problem is tackled as a classification problem, where the significant length of the documents in the provided dataset is addressed by using sentence embeddings in a hierarchical architecture. Three different pre-trained models to generate sentence embeddings have been evaluated and compared: USE, LaBSE and SBERT. Other than sentence embeddings the classifier exploits a bag-of-entities representation of the document, generated using a pre-trained named entity recognizer. The final model is able to simultaneously produce the required date for each subtask.

1 Introduction

To solve the DaDoEval task (Menini et al., 2020) for the EVALITA 2020 competition (Basile et al., 2020) a model should be able to assign a temporal span from a discrete set of candidates to a document, i.e. recognizing when the document was issued. As many other NLP tasks, like author identification or topic assignment, this task can be reduced to a classification problem.

The provided dataset contains documents written by the Italian statesman Alcide De Gasperi in the time span 1901-1954, labeled with the year in which they were issued. The dating task is divided into different subtasks of increasing granularity. The first subtask requires to classify a document into one of five representative periods in De Gasperi's life as identified by historians. (Table 1) The second and the third subtasks require to date a document more precisely, using a five-year span for the former and the precise year for the latter. These subtasks are referred to as the same-genre subtasks.

ID	Period description	Time span
A	Habsburg years	1901-1918
B	Beginning of political activity	1919-1926
C	Internal exile	1927-1942
D	From fascism to the Italian Republic	1943-1947
E	Building the Italian Republic	1948-1954

Table 1: Historical periods of De Gasperi's life

Other than on a blind test set kept from the same-genre dataset, the model has been also evaluated on three additional cross-genre subtasks. In this case, documents coming from a De Gasperi's epistolary archive were used to build an external blind test set. The cross-genre subtasks require to classify documents with the same increasing time granularity as the same-genre ones.

The tasks are evaluated using macro-averaged F1. Baseline results using logistic regression and tf-idf on a bag-of-word representation are provided by the task proponents in table 2.

Subtask	Macro-Average F1
Historical	0.827
Five-years	0.485
Single-year	0.126

Table 2: Proponents baseline

All of the results and the described experiments have been implemented using TensorFlow and executed on the platform Google Colab. The limitations of the platform regarding continuous usage are not negligible and had an acknowledgeable weight in multiple decisions.

In section 2 different approaches to deal with long text classification are described and the various sentence embeddings models are presented. In section 3 the peculiarities of the dataset are discussed. In section 4 the different sentence embeddings models are evaluated and compared with alternative approaches over a single subtask. In section 5 the architecture of the final model used to solve all the subtasks is described, its results are reported in section 6 and discussed in section 7.

2 Methodological survey

The use of pre-trained transformers such as BERT (Devlin et al., 2019) has remarkably improved the state of the art in many NLP tasks, text classification included. Furthermore contextual word embeddings produced by pre-trained transformers are preferable when dealing with polysemy. Documents from a wide time span could manifest lexical change, so polysemy may significantly emerge (Blank, 1999).

When dealing with text classification using the transformer model the first architectural issue is given by the length of the documents. To classify a text a special symbol is usually inserted at the start of the input sequence, then the output corresponding to that symbol is fed into a neural network to retrieve the predicted class. Since the maximum input size for a BERT transformer is 512 tokens, it is unlikely that the whole document will fit. Different architectures are available to overcome this problem.

For certain domains it has been studied that not all of the text is needed to achieve good classification accuracy. For instance Sun et al. (2020) propose to select only part of the text, like the head, or the tail or both, up to reducing the text size to fit the input layer of the transformer. The random selection of tokens inside a document has also proven to be effective for topic classification of academic papers (Liu et al., 2018).

Recently different solutions started to exploit hierarchical architectures, segmenting the text to consequently analyze it in its entirety. The use of sentences may be intuitively perceived as more meaningful than fixed-length segments. Accordingly, three different sentence embeddings solutions have been selected to be implemented and evaluated for the DaDoEval task. All of them provide pre-trained multilingual models, satisfying so the computational constraints and the task requirements.

Sentence-BERT, also known as SBERT, produces sentence embeddings by stacking a pooling layer on the top of a BERT transformer. A pre-trained BERT model is fine-tuned using Siamese networks, back-propagating over the cosine similarity of supposedly semantically related sentences. (Reimers and Gurevych, 2019) A monolingual model can be then distilled and expanded to other languages by training a student model to replicate the behavior of the teacher model, and under the assumption that the vector representation of translated sentences should coincide. (Reimers and Gurevych, 2020). The authors of SBERT published `distiluse-base-multilingual-cased`, a distilled model pre-trained on many languages including Italian.

The Universal Sentence Encoder, or USE, comprises different architectures trained on the same set of tasks to enable transfer learning for many NLP tasks with different requirements. (Cer et al., 2018) The original USE has then been expanded for multilingual applications providing two pre-trained models, a transformer and a CNN, both available on Tensorflow HUB. (Yang et al., 2019)

Lastly, the Language-agnostic BERT Sentence Embedding model, or LaBSE, produces sentence embeddings by using a fine-tuned BERT model. The LaBSE model is designed similarly to SBERT, using two sharing-weights transformers initialized by a pre-trained BERT model. The main difference lies in the datasets and the tasks used for fine-tuning. The authors report the remarkable results of LaBSE for languages unseen but somehow related to those in the training set. (Feng et al., 2020) This result may be useful to fill the gaps between contemporary Italian and the XX-century Italian language in the dataset.

3 Data Analysis

The overall dataset contains 2759 manually labeled documents of variable length written by Alcide De Gasperi during its political life. However, the development dataset provided by the proponents contains only 2210 of them, since the remaining ones are kept for the blind same-genre test set. The dataset is extremely unbalanced since the number of elements per time period varies considerably. For instance by analyzing figure 1 it is evident how some years contribute to the dataset with few documents. The lack of data for these periods remarkably impacts the overall accuracy of the learning process. The development set provided by the proposers has been split into a training set and a validation set to assess the capabilities of the different tested models. The training set was composed by sampling the 80% of the development dataset, leaving the remaining 20% to the

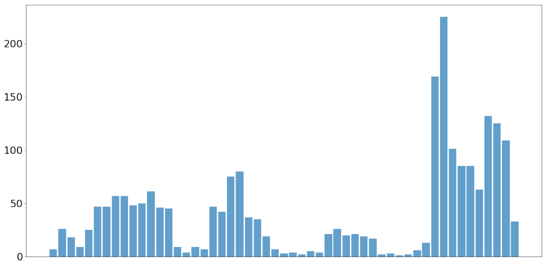

Figure 1: Number of documents per year from 1901 to 1954.

validation split. This choice reflects the proportion between the size of the provided development set and the overall dataset.

Without altering the validation split for the assessment, the training data can be augmented to contrast the unbalancing. The hierarchical solution highly increases the number of tokens that can be used to classify a document, nonetheless the number of sentences per document should be constrained under a fixed constant. When truncating a document to limit the number of sentences, the remaining part is then inserted in the dataset as a new document instead of discarding it. The data augmentation procedure described has been implemented under the assumption that the less represented years contain the longest documents. While this holds for some classes, the effect of data augmentation didn't impact on the overall distribution.

Method	Time
SBERT	223.068s
LaBSE	3364.272s
USE$_{\text{TRANS}}$	154.277s
USE$_{\text{CNN}}$	29.681s

Table 3: Time required by each sentence embedding technique to process the training set.

The tokenizer for the Italian language included in the NLTK library has been used to split each document into a list of sentences (Bird et al., 2009). The content of each sentence has been tokenized instead with a custom tokenizer for each one of the sentence embeddings techniques, since they may require different configurations and their vocabulary must be used. A common issue in this scenario is given by the rate of out-of-vocabulary tokens (Wang et al., 2019), but this hasn't been evaluated since the interfaces offered by the selected models don't offer insights over the OOV rate or other token-level statistics. The time required to produce the embeddings over the training set is reported in table 3.

4 Building blocks selection

Because of the computational limitations, many experiments have been conducted only on one subtask, relegating the others to a subsequent phase. The historical subtask has been chosen because of the better balancing of the dataset and the foreseeable and more promising results. The provided dataset has been split using stratified sampling and data augmentation in a consistent training set and a smaller validation set. The training split covers the 80% of the provided development set, leaving the remaining 20% to the validation one. All of the results are produced by averaging multiple runs, to overcome the non-deterministic and unpredictables effects of the GPUs used for training.

4.1 Truncation based classification

The first experiments used a pre-trained BERT multilingual model for text classification. To overcome the constraint over the input size the documents were truncated up to their first 512 tokens. As expected the truncation has proven to be ineffective since, even after fine-tuning, the model didn't converge on the training set for any subtask. The results aren't significant and therefore not reported.

4.2 Sentence embeddings

Once each document is represented as a sequence of sentence embeddings, two different classification models have been implemented and evaluated. The first is a Recurrent Neural Network with two bidirectional LSTM layers followed by a combination of dropout and dense layers of reducing width. The other classifier is based on the transformer architecture, where a transformer block composed of a multi-headed self-attention layer with 128 heads, dropout and layer normalization is followed by a combination of dropout and dense layers as in the previous solution.

The results of the experiments over the combination of sentence embeddings and the two classifiers are reported in table 4, showing how the combination of SBERT and the transformer-based classifier is the most adequate. With the exception of LaBSE, all the other sentence embeddings models gave better results when coupled with a

	TR			VL		
Top	Loss	Acc	F1	Loss	Acc	F1
LaBSE						
RNN	0.356	0.875	0.884	**0.663**	0.778	0.781
Trans	0.559	0.771	0.697	0.960	0.713	0.616
SBERT						
RNN	0.143	0.955	0.975	0.690	0.824	0.829
Trans	**0.060**	**0.982**	**0.987**	1.235	**0.850**	**0.851**
USE_{CNN}						
RNN	0.193	0.937	0.959	0.780	0.775	0.780
Trans	0.217	0.920	0.937	0.850	0.821	0.819
$\text{USE}_{\text{Transformer}}$						
RNN	0.105	0.969	0.978	0.780	0.815	0.823
Trans	0.192	0.923	0.972	0.773	0.822	0.830

Table 4: Results for the historical periods subtask over training and validation set using different sequence embeddings.

transformer block than with a recurrent neural network. Also, the two variants of USE manifested a more significant gap when coupled with the RNN classifier than with the transformer-based one. Finally, the performance drop of the LaBSE model may reflect a condition also explored by Reimers and Gurevych (2020), where a comparable performance gap with SBERT occurs in semantic textual similarity tasks.

4.3 Bag-of-entities

Another approach to tackle the subtasks consists of exploiting the knowledge of a pre-trained named entity recognizer. It is reasonable to suppose that the entities extracted by a document will produce a good representation for the document itself. In the context of document dating this could be meaningful by assuming that the issues discussed by the author will vary during the years, consequently influencing the entities contained. By building a vocabulary of unique entities it is possible to represent each document as a bag-of-entities, then a multi-layer dense classifier with dropout can be trained to predict the correct time span.

Named entity recognition is achieved using one pre-trained CNN for the Italian language distributed by spaCy (Honnibal and Montani, 2017). Three variants of the same model are provided but, since their differences heavily impact on the model size rather than on the performances (Table 5), the medium sized model has been chosen without further validation. Because of this it is not possible to assess how the performances of the NER alone influence the performances of the overall system.

The NER model returns for each entity a pair containing its content and a label regarding its role. It is possible to consider as a member of the entities vocabulary only the textual content or the unique pair of text and label, both methods were implemented and compared but finally only the label was chosen as representative of the entity.

	Small	Medium	Large
F1	86.57	88.54	89.40
Precision	86.85	88.76	89.56
Recall	86.29	88.33	89.24
Size	13MB	43MB	544MB

Table 5: Model size and benchmark as provided by spaCy for the Italian language pre-trained models. (Explosion.ai, 2020)

4.4 Results

The transformer classifier using sentence embeddings provided by SBERT is chosen as the final candidate since it's the best performing model on the validation set. As previously discussed, the model selection procedure only considered the first subtask because of the magnitude and the balancing of its dataset. To roughly estimate the behavior on all the subtasks both the sentence embeddings classifier and the bag-of-entities solution have been retrained from scratch on the specific subtasks labels and evaluated on the validation set. The results are reported in table 6.

		SBERT+Trans		Bag-of-entities	
Task	Baseline	TR	VL	TR	VL
Historical	0.827	0.930	**0.846**	0.997	0.841
Five-years	0.485	0.482	0.354	0.996	**0.563**
Single-year	0.126	0.086	0.040	0.990	**0.211**

Table 6: Macro-averaged F1 for all the subtasks

5 Model Architecture

It is therefore clear that both the approaches have their advantages on different subtasks. More precisely the sentence embeddings one has proven to be more effective when dealing with the historical periods subtask, while the bag-of-entities obtains better results on the finer ones. The problem of combining these two solutions is now tackled.

The trivial solution would be to hardwire in a single model the different approaches, producing so the output for the first subtask using a sentence embeddings model and for the other subtasks with

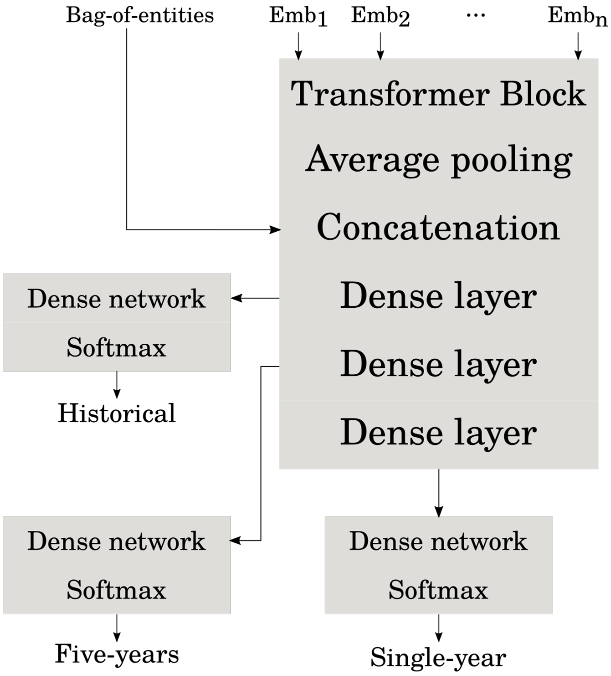

Figure 2: Architecture of the final model.

a bag-of-entities one. While this solution would be acceptable, and seemingly over the baseline according to the estimates on the validation set, it is reasonable to assume that the representations for these subtasks could be shared, improving the performances. Different variations of the same architecture are therefore evaluated on the validation set to monitor such improvement.

In the final model, the sentence embeddings produced by SBERT are fed to a transformer block containing a multi-headed self-attention layer, its output is then averaged and concatenated with the bag-of-entities representation of the document before being fed to a multi-layer neural network. The output of each layer of this network is also fed to a dedicated neural network that produces the output of each subtask. The selected order for the subtasks in the multi-layer dense classifier places the historical classification first, followed by the five-years and then the single-year classification. A graphical representation of the architecture is in figure 2.

Both the reverse of the subtasks order and the absence of hierarchy, by connecting all the classification networks directly to the transformer block, have been tested. Also, the supposed additional value of the concatenation with the entities representation has been experimentally evaluated. The results of these variations are reported in table 7, where the selected final model for the competition is on the fourth row.

		Historical		Five-years		Single-year	
BoE	Order	TR	VL	TR	VL	TR	VL
N	F	0.987	0.828	0.961	0.554	0.577	0.144
N	B	0.988	0.828	0.930	0.566	0.871	0.204
N	A	0.983	0.813	0.973	0.560	0.920	0.228
Y	F	0.991	**0.842**	0.980	**0.599**	0.852	0.236
Y	B	**0.993**	**0.842**	0.988	0.578	0.897	**0.247**
Y	A	0.991	0.820	**0.994**	0.560	0.967	0.242

Table 7: Results for the different subtasks over the training and the validation sets using different architectures. The first column refers to the use of the bag-of-entities representation in the model as in **Y**es or **N**o, the second to the order of the subtasks as in **B**ackward, **F**orward and **A**bsent.

6 Results

The model has been evaluated by using two independent test sets: same-genre and cross-genre. The first one is a blind test set, containing documents from the same source of the provided development dataset. The cross-genre set is instead an external test set, containing documents from a different source, specifically from an archive of epistolary documents of the same subject.

For each subtask two runs per test set were submitted, for brevity in table 8 only the average result of the submitted runs is reported. The model performs over the baseline in the same-genre evaluation for each subtask, also improving the performances with respect to the validation set. Instead, concerning the cross-genre evaluation, the model replicates the results of the baseline and shows a significant drop in respect to the validation set.

		Same-genre		Cross-genre	
	VL	BL	TS	BL	TS
Historical	0.842	0.827	**0.857**	0.368	0.379
Five-years	0.599	0.458	**0.609**	0.171	0.168
Single-year	0.236	0.126	**0.265**	0.020	0.055

Table 8: F1 macro-averaged results for the different subtasks over the validation set (VL), the test sets (TS) and the respective baselines (BL).

7 Conclusions

The contribution of the bag-of-entities representation was certainly helpful, but this should not overshadow the performance improvement given by the introduction of the hierarchical model. The first three rows in the already discussed table 7 report the results of the model without any contribution from the bag-of-entities representation. Whilst neither of these was elected as the best candidate, there is a remarkable improvement over the independent use of the very same building blocks of the final architecture for each subtask.

The described architecture is prone to multiple variations and only some of them have been formally evaluated and compared. Nonetheless, the selected final model was able to surpass the same-genre baseline for all of the different subtasks. Anyhow the performance drop in the cross-genre test should be interpreted as a limit to the generalization power of the chosen model. A wider exploration of the models may increase the overall performances for both the same-genre and the cross-genre tasks.

Also, targeting multiple subtasks at the same time made nontrivial the choice of a final model, therefore it has been carried out intuitively considering the results over the validation set for each subtask. A formal approach to this issue may result in a finer model selection.

Despite the discussed approximations, the use of sentence embeddings models has proven to be effective also on tasks different from the ones they were originally conceived for, and compatible with other representations such as bag-of-entities.

References

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. "O'Reilly Media, Inc.", June. Google-Books-ID: KGIbfiiP1i4C.

Andreas Blank. 1999. Why do new meanings occur? A cognitive typology of the motivations for lexical semantic change. *Historical semantics and cognition*, 61.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *arXiv:1803.11175 [cs]*, April. arXiv: 1803.11175.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May. arXiv: 1810.04805.

Explosion.ai. 2020. Italian · spaCy Models Documentation. https://spacy.io/models/it.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT Sentence Embedding. *arXiv:2007.01852 [cs]*, July. arXiv: 2007.01852.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Liu Liu, Kaile Liu, Zhenghai Cong, Jiali Zhao, Yefei Ji, and Jun He. 2018. Long Length Document Classification by Local Convolutional Feature Aggregation. *Algorithms*, 11(8):109, August. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

Stefano Menini, Giovanni Moretti, Rachele Sprugnoli, and Sara Tonelli. 2020. DaDoEval @ EVALITA 2020: Same-Genre and Cross-Genre Dating of Historical Documents. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]*, August. arXiv: 1908.10084.

Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. *arXiv:2004.09813 [cs]*, April. arXiv: 2004.09813.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to Fine-Tune BERT for Text Classification? *arXiv:1905.05583 [cs]*, February. arXiv: 1905.05583.

Hai Wang, Dian Yu, Kai Sun, Janshu Chen, and Dong Yu. 2019. Improving Pre-Trained Multilingual Models with Vocabulary Expansion. *arXiv:1909.12440 [cs]*, September. arXiv: 1909.12440.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual Universal Sentence Encoder for Semantic Retrieval. *arXiv:1907.04307 [cs]*, July. arXiv: 1907.04307.