Extracting structured metadata from multilingual textual descriptions in the domain of silk heritage

M.Besher Massri
Jožef Stefan Institute, Slovenia
besher.massri@ijs.si

Dunja Mladenić
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

In this paper, we present a methodology for extracting structured metadata from museum artifacts in the field of silk heritage. The main challenge was to train on a relatively small and noisy data corpus with highly imbalanced class distribution by utilizing a variety of machine learning techniques. We have evaluated the proposed approach on real-world data from five museums, two English, two Spanish, and one French. The experimental results show that in our setting using traditional machine learning algorithms such as Support Vector Machines gives comparable and in some cases better results than multilingual deep learning algorithms. The study presents an effective approach for categorization of text described artifacts in a niche domain with scarce data resources.

KEYWORDS

Information extraction, Text classification, Silk heritage, Transformers, Support Vector Machines.

1 INTRODUCTION

When looking to improve the understanding of silk heritage we find that the data available in the museums often lack semantic information on the artifacts or have them to some extent included in textual descriptions. To facilitate automatic analysis of silk heritage data and support digital modeling of the weaving techniques, we propose multilingual metadata extraction from textual descriptions provided by the museums.

We propose the usage of machine learning techniques to model the target variables, referred here as slots to align with the terminology of information extraction. Using machine learning methods we build a model for each of the target variables in order to annotate the text. This enabled us to add metadata to the silk heritage artifacts of the museums. The domain experts collaborating on Silknow project [9] have identified four kinds of metadata information that would be useful and are contained in texts of at least some of the targeted museums. We treat these as four slots for information extraction, where the list of possible slot values for each of the four was defined by the domain experts. Based on that we formed a multi-class dataset for each slot.

The corpora of text included were in three different languages (English, Spanish, and French) from five different museums, with a total of 500 museum records used in the study. After the data processing and annotation, we generated 24 binary datasets and 19 multi-class datasets (four for English, two for Spanish, and one for French). Using machine learning techniques we trained classifiers on the labeled data examples to predict the labels (slot values) based on the textual descriptions. Despite relatively small and unbalanced data corpora, using sampling techniques and weighted loss function helped mitigate the issue. In an experimental evaluation, we observed that on our data using traditional methods might be as good as using deep learning models when the data is scarce. However, using deep learning allows for building multilingual models that scale across different languages.

The main contribution of this paper is in proposing an approach to adding metadata to historical artifacts based on applying machine learning on multilingual textual descriptions of the artifacts. Moreover, we have defined the learning problem in collaboration with domain experts and performed evaluations on real-world data in English, Spanish, and French. The rest of this paper is structured as follows. Section 2 provides a description of the data, Section 3 describes the proposed methodology, Section 4 gives the results of the evaluation and Section 5 concludes the paper summarizing the approach and the findings.

2 DESCRIPTION OF DATA

We used the SilkNow knowledge graph [8] as our source of data. The source consists of records of different museums in different languages as shown in Table 1. The largest are MET with8364 artifacts in English, VAM with 7231 artifacts in English, and Imatex with 6799 artifacts in Spanish. We have used a subset of the data that contain artifacts with provided metadata and textual descriptions in related fields that were pointed out as relevant by the domain experts. Each record consists of the basic information about the object, such as the title and the museum it belongs to, along with two other sets of attributes, textual attributes, and categorical attributes. Textual attributes hold a textual description of the object in several fields, such as physical description and a technical description. The categorical description holds metadata information, such as technique or materials used. However, the data quality varies across the museums and records. Some museums are rich in both textual and categorical attributes, like the VAM museum, and others have short/low-quality textual attributes like Imatex. Also, some records have a text description in their categorical attributes instead of a single category value.

The metadata fields that we have considered are weaving technique, weave, motifs, and style. The list of labels or slot values for each of the metadata field (i.e. slot for information extraction) were compiled by the domain experts. These values describe the silk artifacts' nature and structure. Each of those slot values is represented by a term and a list of alternatives, up to four alternatives per term. Examples of slot values are satin, twill, and tabby, representing possible values of the weave slot.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Information society '20, October 5–9, 2020, Ljubljana, Slovenia
© 2020 Association for Computing Machinery.

Museum	Language	Count
CER	Spanish	1296
Garin	Spanish	3101
Imatex	Spanish	6799
Joconde	French	376
MAD	French	763
MET	English	8364
MFA	English	3297
MTMAD	French	663
RISD	English	3338
VAM	English	7231

Table 1: Museums from the Silknow knowledge graph showing the language of the artifacts and the number of artifacts included in the knowledge graph.

3 METHODOLOGY

3.1 Annotating datasets with slot values

Based on the data and target variables, two types of datasets were formed for two types of text classification tasks. The first type is binary classification dataset, in which the target class is one of the slot values. The other is multi-class classification dataset, in which a dataset is formed for each of the four slots in each museum, where the target classes are the slot values that fall under the selected slot in addition to extra "other" class indicating that the example doesn't fall under any of them.

For forming the binary classification dataset we used a simple string matching approach. For each target class in each museum, examples were formed out of textual attributes of the museum records that contain a mention of either one of the possible value terms or its alternatives. Categorical attributes of the same record were used to determine the label of the example. The task is to classify whether the example has the slot value against the other slot values of the same slot. Each item is classified as True if the categorical attributes contain only the target value or one of its alternatives but not any of the other slot values' terms or their alternatives. If there is no mention of the slot value term or alternatives, then it's classified as false. If it contains this slot value' term along with other slot values' terms then it's considered as indeterminate and the example is removed.

To form the multi-class datasets, we merged the datasets of the same museum with target classes representing slot values that fall under the same slot. The true items of each slot value dataset formed the set of the examples with that slot value as the labels. The items that are false in each slot value dataset formed the "Other" class in the multi-class dataset.

3.2 Binary Classification Tasks

For binary classification, we used TFIDF word-vector representation for generating the feature vectors and trained a Linear Support Vector Machines (SVM) as the classifier using scikit-learn library [5]. All dataset were split into train and test using 80-20 stratified split. We performed a grid search with 5-fold cross validation on the training part using the following options:

- stemming, lemmatisation, or none
- max document frequency: [0.95.1.0]
- min document frequency: [0,0.05]
- SVM tolerance: [1e-4,1e-5]

The features were generated from sequences of words, referred to as n-grams, of length 1, 2, and 3. The remaining parameters were left unchanged from their default values. We used nltk [1] library for tokenization, SpaCy [4] for lemmatization, and Snow Ball Stemmer [6] for stemming.

Due to the methodology of data labeling, we sometimes ended up with a highly imbalanced datasets having a lot more negatives than positives. Therefore, in the binary dataset, we took a random subset from the negative examples to match the positive count. In addition, some examples were generated from the same records, by having more than one textual record with mentions of the same class's term/alternatives, therefore, corrections have been applied to the dataset by putting all examples of the same record in either train or test but not in both. This process was done to ensure no leakage occurs by potentially having highly similar textual text in train and text.

3.3 Multi-class Classification Tasks

For multi-class classification, we used a deep learning approach. The architecture consists of a pre-trained transformer, an LSTM layer, a dropout layer, a dense (linear) layer, and finally a soft-max activation layer. For the transformer we used BERT [3], multi-lingual BERT, and XLM-ROBERTA [2]. The loss function used was a cross-entropy loss with Adam as the optimizer. We used PyTorch framework [7] and hugging-face transformers library [10].

Considering that some of the datasets have a large class imbalance, which can be a couple of thousand examples of the majority class and only a few examples of the minority classes, we experimented with several class-weighting schemas. First, we tried assigning weights to the classes in the loss function is inversely proportional to the number of examples of each class. In addition, when we used weighted sampling with return for loading the examples into batches. This had the effect of over-sampling the minority classes and under-sampling the majority classes to achieve as balanced batch representation as possible. Finally, we tried a derivable version of F1 Macro as a loss function where the prediction matrix is taken as a probability rather than a binary value.

4 RESULTS

4.1 Experimental Datasets

The dataset collection methodology was applied to 10 museums and 4 categories holding more than 150 class values overall. However, most of the datasets have no positive items. In this research, we have selected datasets with at least 10 positive examples for binary classification tasks and at least 10 non-other in multi-class tasks. This final list consists of 24 binary datasets and 19 multi-class datasets. These datasets are used for training machine learning classifiers.

4.2 Binary Classification Tasks

For binary Classification, we applied the described methodology on all the datasets with at least 10 positive examples. The results of binary classification are consolidated in Table 2.

The graph in figure 1 displaying the correlation between the number of examples and the F1 score reveals a weak correlation of 0.19. We can see that when having more than 600 examples, we achieve F1 over 0.8. Upon closer inspection on the museum level, we found that the best results are achieved in the MFA museum on motifs and weaving technique and Joconde museums on weave.

Museum	Slot value	Slot	Language	#Exs	Accuracy	Precision	Recall	F1
cer	bordado	weaving technique	Spanish	278	0.89	0.87	0.93	0.9
cer	motivo vegetal	motifs	Spanish	146	0.57	0.56	0.6	0.58
cer	tafetÃ¡n	weave	Spanish	581	0.77	0.9	0.6	0.72
cer	terciopelo	weaving technique	Spanish	118	0.67	0.67	0.67	0.67
garin	brocatel	weaving technique	Spanish	932	0.88	0.85	0.92	0.89
garin	damasco	weaving technique	Spanish	1748	0.9	0.92	0.87	0.89
garin	espolÃn	weaving technique	Spanish	972	0.88	0.89	0.88	0.88
joconde	Satin	weave	French	159	0.91	0.9	0.95	**0.93**
joconde	Taffetas	weave	French	110	0.95	0.92	1	**0.96**
mfa	Lace	motifs	English	190	0.92	0.9	0.95	**0.92**
mfa	plain	weaving technique	English	130	1.00	1.00	1.00	**1.00**
vam	brocade	weaving technique	English	634	0.87	0.87	0.87	0.87
vam	damask	weaving technique	English	480	0.84	0.85	0.83	0.84
vam	Ear	motifs	English	262	0.83	0.84	0.81	0.82
vam	Edge	motifs	English	178	0.81	0.87	0.72	0.79
vam	embroidery	weaving technique	English	1614	0.85	0.86	0.83	0.84

Table 2: Results for the binary classification task.

Overall the best results are achieved by MFA and Joconde with an average F1 of .96 and .95 respectively followed by Garin, VAM, and CER with the average F1 of .89, .81, and .72 respectively.

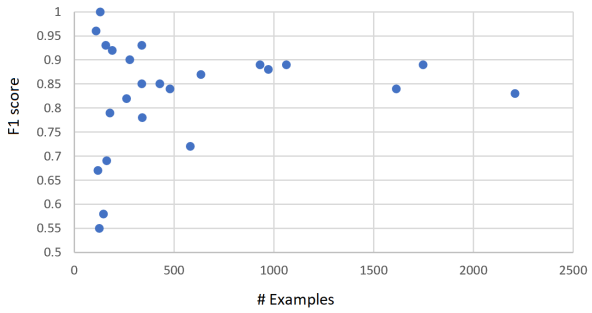

Figure 1: F1 score vs #Examples showing good performance on the largest datasets, when the number of examples is at least 600.

4.3 Multi Class Classification Class

4.3.1 Use Case: Detecting Weave Slot from VAM museum. We selected the VAM Weave slot as a use case dataset to perform hyperparameter tuning and select the best configurations for weighting. The dataset contains 2760 items with a baseline of 52.9% distributed across 4 classes: Satin, Tabby, Twill, and Other. The dataset was split into train, test, and validation in the form of 60-20-20 split. The results in Table 3 show that using class weighting in both loss function and sampling provides the best results w.r.t both classification accuracy and F1. Using F1 as a loss function sometimes provided good results but was discarded as it was not stable across different datasets. In addition, decreasing the learning rate improved results and stabilized the training curve. Finally, using the XLM-ROBERTA transformer showed an improvement in accuracy. The number of epochs was determined based on the accuracy performance of the validation dataset. The training would stop when the accuracy did not improve for the last 15 epochs. The accuracy (F1 micro) was chosen over F1 macro because of the large fluctuation in F1 macro value across training epochs caused by having minority classes with few examples.

Model configuration	Accuracy	F1
Base model	84.6	43.1
Weighted loss	82.1	47.2
Weighted sampling	82.6	52.2
F1 loss function	77.5	59.1
weighted sampling and f1 loss	52	22.8
Weighted loss and weighted sampling	84.8	54.7
+ Learning rate 1e-4 $\longrightarrow$ $5e-6$	86.1	57.9
Multi-Lingual BERT	85.3	55.2
XLM-ROBERTA	87.5	53.6

Table 3: Comparison between different model configuration on the Weave Slot detection in VAM Dataset

Comparing the learning curves of BERT and multi-lingual BERT in figure 2 reveals that despite the comparable results, the multi-lingual BERT took double the number of epochs to stabilize and finish training compared to its BERT counterpart. This can be due to the fact that Multi-lingual BERT is trained in many languages and it needs more fine-tuning to adapt to any certain language, whereas the BERT transformer was trained in English-only documents.

4.3.2 Generalizing towards all datasets. After we experimented with different parameter settings, we decided to use the following parameters on all the datasets: Weighted Loss function and Weighted Sampling for batches; learning rate of $5 * 10^{-6}$; batch size of 16 for BERT and 12 for multi-lingual BERT and XLM-ROBERTA, due to memory limits; 1024 Units for LSTM Layer; dropout layer of 0.5.

Moreover, the datasets were tested against three types of transformer: Language-Specific BERT, Multilingual BERT, and XLM-ROBERTA, as well as the SVM classifier. The accuracy results in Table 4 show that on most of the datasets SVM performs better or comparable to the deep learning models.

Museum	Lang	Slot	Baseline	# Cls	# Exs	SVM	BERT	Multi BERT	XLM-ROBERTA
VAM	English	Weave	52.9	4	2760	82.8	86	85.3	**87.5**
VAM	English	Weaving Technique	35.9	14	3525	77.6	**80.1**	78	78
VAM	English	Motifs	84.8	9	5500	**91**	**90.6**	87.4	87
CER	Spanish	Weave	59.3	5	945	**75.1**	**75.1**	64	72
CER	Spanish	Weaving Technique	61.1	11	720	**74.3**	**74.1**	71.5	66
Joconde	French	Weave	55.6	4	180	66.7	30.6	86.1	**91.7**
Joconde	French	Weaving Technique	60	5	150	**97.2**	70	76.7	63.3

Table 4: Results for the multi-class classification task.

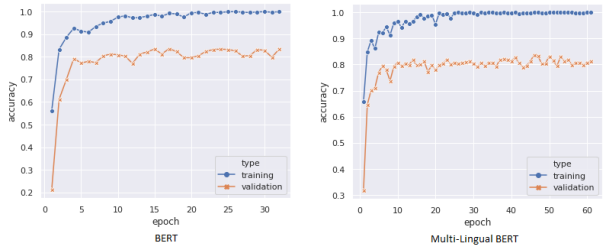

Figure 2: Comparison of a learning curve between BERT and Multi-Lingual BERT as a transformer in the deep learning model trained on the VAM museum Weave Slot dataset.

5 CONCLUSION AND FUTURE WORK

We propose an approach to extracting metadata from a multilingual text description of silk heritage domain museum artifacts. The datasets had several specifics that made the model development a non-trivial task. First, the size of the dataset sometimes was too small to train a model. Second, some class values have considerably more examples than others, which caused the datasets to be imbalanced. Finally, in the preparation phase, the datasets were labeled to accommodate the described issues, which in itself is an approximation and carries an inherent error rate. We have improved the performance of the model by oversampling minority classes, under-sampling majority classes, and using a class-weighted loss function. In addition, by performing cross-validation in the binary classification case or adding a dropout layer and validating based on a validation dataset, we managed to mitigate some of the over-fitting behavior caused by having a little amount of data. We believe that the over-fitting could be mitigated further by using regularization on the LSTM layer, as well as using weight-decaying in the optimizer.

The experimental results show that with low data quality and having not enough data, traditional methods such as SVM in some cases outperform deep neural network models. We expect that the results could be improved by having an assembly of those models instead of using one of them only, which is a part of the future work. Furthermore, one can fine-tune each model independently to achieve better performance.

In future work, we plan to test cross-museum learning by training on one museum and predicting other museums both in the same language and in different languages using multi-lingual transformers. This has practical value for labeling the data in the museums that do not contain metadata information but do have suitable textual descriptions of the artifacts.

ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and SilkNow European Unions Horizon 2020 project under grant agreement No 769504.

REFERENCES

[1] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, (July 2020), 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. https://www.aclweb.org/anthology/2020.acl-main.747.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[4] Matthew Honnibal and Ines Montani. spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, (2017).

[5] 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.

[6] Martin F. Porter. 2001. Snowball: a language for stemming algorithms. Published online. Accessed 11.03.2008, 15.00h. (2001). http://snowball.tartarus.org/texts/introduction.html.

[7] [n. d.] Pytorch: an imperative style, high-performance deep learning library. In.

[8] 2020. Silknow knowledge graph data. https://github.com/silknow/converter/tree/master/output. (2020).

[9] 2020. SilkNow project. https://silknow.eu/. (2020).

[10] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. [n. d.] Huggingface's transformers: state-of-the-art natural language processing.