Labelling the past: data set creation and multi-label classification of Dutch archaeological excavation reports

Brandsen, A.; Koole, M.

Lang Resources & Evaluation
https://doi.org/10.1007/s10579-021-09552-6

ORIGINAL PAPER

Labelling the past: data set creation and multi-label classification of Dutch archaeological excavation reports

Alex Brandsen[1] · **Martin Koole**[2]

Accepted: 2 July 2021

Abstract The extraction of information from Dutch archaeological grey literature has recently been investigated by the AGNES project. AGNES aims to disclose relevant information by means of a web search engine, to enable researchers to search through excavation reports. In this paper, we focus on the multi-labelling of archaeological excavation reports with time periods and site types, and provide a manually labelled reference set to this end. We propose a series of approaches, pre-processing methods, and various modifications of the training set to address the often low quality of both texts and labels. We find that despite those issues, our proposed methods lead to promising results.

Keywords Multi-label classification · Grey literature · Machine Learning · Archaeology

1 Introduction

Over the past decades, the archaeological domain has produced a large quantity of literature in the form of excavation reports, scholarly articles, and books. The Archaeological Grey literature Named Entity Search (AGNES) project (Brandsen et al., 2019) aims to uncover any relevant information from Dutch archaeological

✉ Alex Brandsen
a.brandsen@arch.leidenuniv.nl

Martin Koole
martin_koole@live.nl

[1] Faculty of Archaeology, Einsteinweg 2, 2333CC Leiden, The Netherlands

[2] Leiden Institute of Advanced Computer Science, Niels Bohrweg 1, 2333CA Leiden, The Netherlands

Published online: 28 July 2021

excavation reports. Such reports are often grey literature: material that is either unpublished, or published in a non-traditional manner. Information uncovered by AGNES will be made easily accessible through a specifically designed search engine, enabling researchers to search for relevant texts.

In this search engine, certain aspects of documents are used for faceted search, allowing archaeologists to filter search results on site type and time period metadata fields. This information need is further detailed by Brandsen et al. (2019). AGNES currently only indexes documents with manually assigned metadata, but in the near future, documents without metadata will be added. To allow for faceted search on these documents as well, we propose to automatically assign metadata. Manual labelling is an unfeasible task due to the amount of texts: there are currently an estimated 70,000 documents and four to five thousand are added each year. Due to this volume, using text mining and machine learning techniques becomes a necessity.

In this paper, the labelling of Dutch archaeological excavation reports with time periods and site types[1] will be addressed in the form of a multi-label classification task.

We first create a manually labelled reference set, and then define a collection of pre-processing steps, classification methods, further text formatting and sampling techniques that lead to a multitude of different combinations. We determine which approaches are suitable for this particular type of data, and we discuss how these methods could be further improved.

Although reports are typically freely available in online repositories and archives, processing the documents proves to be rather difficult for four main reasons:

1. Some of the documents are scanned hard copies, and the OCR process introduces noise
2. The documents are only available in PDF format, and conversion to plain text introduces noise
3. The training data labels are derived from the metadata values which has been added through a free text field, leading to highly diverse and inaccurate metadata
4. There is a large number of target labels (146 site types, 42 time periods) with a strong class imbalance

See Table 1 for examples of point 1 to 3, and see Figs. 2 and 3 for point 4.

Besides being useful for faceted search, this machine learning approach can also be helpful for document depositors when they assign metadata to new documents, by suggesting a number of possible labels for the user to choose from. If implemented, this will also lead to more structured metadata in the future, as it prevents free text input on these fields. With these goals in mind, we address the following research questions:

[1] *Complextype* in Dutch. This can be regarded as a 'subject' field, a site type is what type of past human behaviour has been encountered. Some examples include settlements, churches, graves, etc.

Table 1 Examples of noise introduced by (1) OCR mistakes, (2) PDF to text conversion and (3) manual metadata entry in free text fields (locations in time period field)

	Error	Correct
1	IJsertijdbewoning	IJzertijdbewoning
2	H et huidige landschapsbeeld	Het huidige landschapsbeeld
3	Time periods: Gelderland, Ede, Nieuwste Tijd	Time periods: Nieuwe Tijd

Errors are underlined

- Which combination(s) of text pre-processing steps, data augmentation/balancing, document pre-selection, and classification method yields the highest F1 scores?
- Are the best combinations the same across the different categories and labels, or do specialised combinations per category lead to better results?
- To what extent can we classify Dutch excavation reports into time periods and site types?

While multi-label classification is a well-studied subject, in this paper we perform this task on a noisy data set in an expert domain, making the process more challenging. Even though the difficulty of the task is high, we achieve decent results: we achieve comparable or better scores when compared to similar studies in other domains (Golub et al., 2020; Kleppe et al., 2019). We also specifically test which pre-processing methods have a positive effect on classification, and provide the created data in an online repository[2].

2 Related work

2.1 Text mining in the archaeological domain

Vlachidis and Tudhope (2012) address the semantic annotation of English archaeological documents, a process similar to our classification task. Despite a difference in language, highly similar issues are found in the data set for example. These include the extraction of relevant document sections, scarcity of vocabulary resources, and the construction of a reference set in order to assess the results. Vlachidis and Tudhope (2012) also address the issues of this type of (grey) literature in general. Often, specific archaeological items or names will be mentioned within texts, but hold barely any relevance to the overall topic. Similarly, a variety of terms, such as ‘context’, ‘deposit’ and ‘cut’ yield specific archaeological definitions, but would normally often be seen as common, and therefore not meaningful.

Like our own study, the Archeotools project Jeffrey et al. (2009) also aimed to automatically generate metadata for faceted search. They focused on ‘What’,

[2] https://doi.org/10.5281/zenodo.3676703.

'Where' and 'When' facets. However, they considered this to be an information extraction task instead of a classification task. As such, they have a slightly different approach based on Named Entity Recognition (NER). The extracted entities are then matched to entries in a English archaeology thesaurus to provide structured metadata. The OPTIMA system by Vlachidis and Tudhope (2016) also focuses on information extraction, but using hand-crafted rules instead of machine learning.

In Dutch, no document classification seems to have been done, but some researchers have experimented with NER, like Paijmans and Brandsen's research on detecting time periods (Paijmans & Brandsen, 2010), Vlachidis et al. with their work in the ARIADNE project (Vlachidis et al., 2017) and the more recent work by Brandsen et al. (2019, 2020). In the broader context of cultural heritage (also including museums, monuments, etc), Sporleder (2010) gives an overview of the use of Natural Language Processing (NLP) in this domain, but there is a focus on information extraction, not document classification. In an even broader context, Fiorucci et al. provide a summary of—and a critical reflection on—the use of machine learning in the cultural heritage sector, but do not address NLP in any detail (Fiorucci et al., 2020).

2.2 Multi-label text classification

As already mentioned in the introduction, the classification of Dutch archaeological reports is a multi-label classification problem with many categories and a large class imbalance, as illustrated by Figs. 2 and 3. These characteristics are not unique to the archaeology domain, and are also often encountered in e.g. the biomedical domain (Laza et al., 2011) and library domain (Golub et al., 2020).

A multi-label classification problem refers to a set of items which can be assigned zero or more labels, according to defined categories. As opposed to binary classification, where an item can have one of two labels, i.e., true or false. Multi-class classification shares the multitude of categories, but here, each item receives one label, rather than zero or more.

Cherman et al. (2011) present a case study for multi-label classification with many categories. They propose to transform the n-label problem to n binary relevance problems. One major advantage is that the computational complexity is drastically lowered compared to other multi-label strategies. A disadvantage however, is that relationships between labels cannot be taken into account. In our case, this is not likely to be a problem: though consecutive time periods are naturally more likely to occur together, there are no direct relationships between these periods in terms of archaeological finds. As a matter of fact, time periods are generally defined based on finds, or the material culture of people in the past (Renfrew & Bahn, 2019). Because of this principle, we decided not to introduce a smaller penalty for consecutive periods compared to periods that have a (large) time span between them, i.e., ordinal evaluation. Thus, similarly to the site types, we consider the evaluation of the time periods to be discrete.

To evaluate our methods, we use the F1 score, which is the weighted average—or harmonic mean—of the *precision* and *recall*. Precision is defined as the fraction of positive items that are predicted correctly, and recall is the fraction of positive items

retrieved with respect to all positive items within the set (Powers, 2011). As the harmonic mean over these values, the F1-score is defined as follows:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Due to the nature of the task, there is no preference for either recall or precision, and as such we do not use the more recall oriented F2 score, or the more precision oriented F0.5 score (Sasaki, 2007).

With regard to the class imbalance, Joachims (1998) showed the robustness of Support Vector Machines (SVM), as they provide built-in protection for unbalanced data sets. Another promising approach is the integration of Doc2Vec, a neural network that converts texts into vector representations. In combination with an SVM, Doc2Vec yields high results in terms of *F1 scores* on the task of multi-labelling, for example in ground lease documents (de Romas, 2019).

Finally, a recent state-of-the-art classification technique is the Bidirectional Encoder Representations from Transformers (BERT) architecture (Devlin et al., 2018). This method distinguishes itself from traditional sparse word vectors by learning pre-trained dense language representations from unlabelled data, creating context sensitive embeddings. As such, BERT yields a better contextual understanding of languages, and can lead to improved performance on a lot of NLP tasks.

3 Data

In this section, we discuss and analyse the raw data. First, a general description of the data set will be provided based on document titles, content observations and relevant statistical properties. Next, we present the method that has been used in order to extract labels from available metadata, to construct the training and test sets. We then create an overview of the categories extracted from the data and the corresponding labels based on the *Archaeologisch Basis Register* (ABR) notation, further detailed in Sect. 3.2. Finally, observations are made regarding the difficulties that the data set might introduce in later stages of the overall research process.

3.1 Source data

We use all documents in the 'archaeology' category in the 2016 version of the Data Archiving and Networked Services (DANS) repository, one of the largest Dutch e-depots. This data set consists of just over 65,000 files, all of which are in PDF format. Examples of included files—based on document titles—are (excavation) reports, publications, separate appendices and figures, letters, and metadata. Although we have not statistically tested the representativeness of this data set, it represents almost all the output of commercial archaeology units from the last 30 years or so, spanning all time periods, site types and different types of reports.

Quite often reports have been split into multiple PDFs, one file for each chapter and appendix is quite common for longer reports. For our research, AGNES already provides a collection in which all files have been converted to both XML and raw text format, which allows for the use of information retrieval and text classification. In this research, we only use the raw text files, which have been created using the `pdftotext` software (Glyph & Cog LLC: pdftotext, 1996).

We see that the conversion of the PDF files to the required text format introduced a lot of noise. This includes headers, page numbering and various indices appearing at random positions in the text. The main culprits are tables and figures, which are no longer recognisable after conversion. Brandsen et al. (2019) estimate that around 15% of all documents are OCRed, a process likely to introduce noise even before the PDF to text conversion. Luckily, this percentage will only decrease, as more and more born digital documents are added over time.

3.2 ABR ontology

The ABR is a Dutch archaeological ontology describing time periods, artefacts, materials and site types, and their corresponding shorthand codes, created and maintained by the RCE (*Rijksdienst voor Cultureel Erfgoed*, the Dutch heritage agency) (Brandt et al., 1992)[3]. The main aim of this ontology is to provide an exhaustive list of terms and definitions for terms commonly used in archaeology as a reference.

Unfortunately, the ontology is not geared towards NLP, as concepts are often defined in ways that do not mirror their use in running text, e.g. the entry for 'perforated axe' is '*bijl, doorboord*' (axe, perforated). Also, synonyms and lemmas/stems are not included, and terms might occur in multiple categories (e.g. 'Iron' as a material, or part of the time period Iron Age). While this does not pose a problem for creating a set of target labels for machine learning (as described in the next section), we are aware that this will cause noise in the term extraction described in Sect. 4.5, where we use entities as features in a classifier.

3.3 Definition of categories

Classification is to be done in two dimensions: time periods and site types. The categories for time periods and site types are based on the ABR ontology. These codes are specifically defined for the description of Dutch archaeological concepts. In general, the ontology will provide us with a thesaurus, linking aforementioned codes, textual representations and corresponding descriptions. Furthermore, the ontology introduces sub-categorisation for both time periods and site types. Tables 2a and b show an overview of the categories we will take into account.

Ideally, we would also like to label the documents on artefacts (objects, e.g. an axe) and materials (e.g. flint), as these categories, combined with site type and time period, are the most used aspects in the information needs of archaeologists (Brandsen et al., 2021). Unfortunately, this is currently not possible as we do not

[3] Available online at https://thesaurus.cultureelerfgoed.nl/.

Table 2 Overview of the included labels, full names and the number of sub-categories for each main category in time periods and site types

(a) An overview of the eight time period categories and number of sub-categories

Time periods		
Label	Category	Sub-categories
paleo	Paleolithic	5
meso	Mesolitic	3
neo	Neolithic	9
brons	Bronze Age	5
ijz	Iron Age	3
rom	Roman Time	9
xme	Middle Ages	8
nt	Modern	3

(b) An overview of the eleven site type categories and number of sub-categories

Site types		
Label	Category	Sub-categories
xxx	Unknown	1
cthd	Cult/sanctuary	8
bewv	Habitation/settlement	32
apvv	Agricultural production	12
wrak	Shipwreck	3
idnh	Industry	21
sv	Shipping	8
gw	Resource extraction	9
bgr	Grave field	1
bgv	Burial (general)	17
infr	Infrastructure	25

Category names are translated from Dutch

have training data for these fields, because this information was not recorded for our training set.

3.4 Obtaining the document labels from the data

As mentioned briefly in the introduction, the data set has associated metadata for each document, as entered by the document authors at time of deposition in the DANS archive. The metadata entry was originally performed through a free text field, but has since been updated to dropdown boxes with specified ABR codes, and they are not required fields. Instructions for metadata entry are available on a separate page. Due to these factors, we see that the quality is relatively low: many

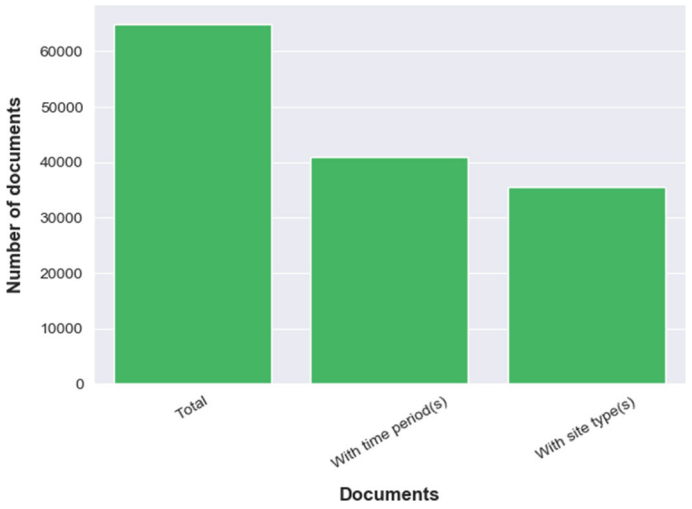

Fig. 1 The number of documents and available metadata values

documents are missing metadata, there are large inconsistencies between documents, and we even encountered wrongly entered metadata. To create a training set for document classification, we retrieve the manual metadata and clean it where possible, which is described below.

The retrieval of manually assigned metadata (time periods and site types) for each document is done by means of an XML crawler that uses the DANS Easy API.[4] All fields can have zero or more entries.

3.5 Exploration of the extracted labels

We encountered several issues with the retrieved metadata values. First, there are over 1200 and 2600 unique metadata values retrieved via the XML crawler for the time periods and the site types respectively. Some of these metadata values are valid, but as stated in Sect. 3.3, we will only include a predefined selection of labels. Many other metadata values are simply not documented in the ABR ontology, instead being variations or older versions of actual labels, erroneously spelled labels, or completely irrelevant: for example names of cities instead of time periods. This reoccurring issue is because metadata was originally entered in a free text field where mistakes can be easily made. In Sect. 3.6 we describe how we processed the extracted metadata values into the set of predefined ABR labels set which we can use for classifier training.

Overall, more than 24,000 files do not have any metadata for the included time periods, and 29,500 files have no site type metadata (see Fig. 1).

[4] https://easy.dans.knaw.nl/ui/home/.

3.6 Pre-processing the metadata

In order to introduce consistency, we convert all metadata values to a single, general format that only includes valid labels in the form of ABR codes. However, for time periods alone, over 1200 unique metadata values first have to be mapped onto the 45 labels (or 53 including main categories) we actually take into account. This process was done automatically where possible, but still required manual inspection and decision making regarding unclear metadata. This means that some unwanted labels are assigned to files, further affecting the classification process. In combination with the presence of erroneously assigned labels—those of correct ABR format, but simply not reflecting the content of the document—the training set will inevitably contain an unknown percentage of incorrect labels.

This will most likely harm the performance of the models to some extent, but without manually labelling a large amount of documents as a training set, it would be impossible to overcome this problem. For the test set, we do create a manually labelled set (see Sect. 4.4), so we can evaluate the performance even with a noisy training set.

For the site types, there were approximately 2600 unique values in the retrieved metadata. Due to the high number of included categories—11 main, 146 in total—we opted to only map labels in outdated ABR notation to current ones, and check for textual formats and their plural forms. Here, no further exhaustive manual labelling was done as the amount of metadata values and target labels is too large, making manual labelling too time consuming. Similarly to labelling the time periods, valid ABR codes might be erroneously assigned to documents, again decreasing the reliability of the training set.

After parsing the metadata for time periods to a valid ABR based format, we define the following rules to assign additional categories as to further introduce consistency in terms of time span:

- Whenever a file is labelled with a category of the lowest hierarchical level, all parental categories will be assigned as well. For example, when a file is only labelled by *lmea* (Late Medieval A), then this file will be given additional labels *lme* (Late Medieval) and *xme* (Medieval—main category).
- When a file is only labelled with an intermediate level category, for example *lme*, its parental category will be assigned, *xme*, and its child categories, *lmea* and *lmeb*.
- When a file is labelled only with a main category, then *all* child categories from all hierarchical lower levels will be assigned as well.

We are aware that the last two rules are based on the following assumption: when someone labels a document as a top level time span (e.g. Medieval), they mean that items from the entirety of the Medieval period have been found, so from early to late Medieval. However, in some cases this will not hold true, as archaeologists often find items that can only be broadly defined as e.g. Medieval, and it is not clear from which of the sub-periods the item originates. Again, this will introduce some noise in the labels, as we cannot with certainty predict which sub-periods are

Table 3 Examples showing the conversion of free text metadata entries to structured label codes

Assigned metadata	Extracted labels	Type of conversion
ABR:NT	nt, nta, ntb, ntc	Sub-categories added
Late Middeleeuwen en Nieuwe Tijd	xme, lme, lmea, lmeb,nt, nta, ntb, ntc	Free text to label codes, with sub-categories
Gelderland; Ede;	None	Wrong metadata (location), no label assigned
Dijken,rivierduinen, prospectie, terpen	infr, infr.dij, bewv, bewv.tw	Free text to label codes, with main categories; only two out of four terms are valid ABR codes

actually present, but we still feel this is the most consistent way to generate our labelled data set.

For site types, there are only two levels of hierarchy. We will therefore limit the addition of categories to only main categories in cases where these are not yet included when only a sub-category is provided. When only a main category is present however, we will not assign any additional sub-categories, as the exact site type(s) cannot be derived.

After this process, we end up with an average of 8.1 labels per document (median: 4, max: 53) for time periods, and an average of 1.65 (median: 0, max: 18) for site types. Table 3 shows some examples of manually assigned metadata, and which labels were extracted after the pre-processing steps described above.

4 Methods

In this section, we describe how we pre-processed the documents, modified the training set, constructed a manually labelled reference set, and selected the classification models.

4.1 Document pre-processing

In order to prepare the textual data for classification tasks, we define several pre-processing methods, some of which are specifically targeting characteristics of observed noise, such as an abundance of punctuation or other non-alphabetical marks. Pre-processing steps include:

1. Lower-casing
2. Removal of all punctuation marks
3. Removal of abundant spacing
4. Removal of digits
5. Removal of all non-alphabetical marks
6. Stemming by means of NLTK's Snowball Stemmer[5] for Dutch words
7. Removal of tokens with a length equal to or less than three
8. Removal of stop words

We define ten combinations of these pre-processing steps, to find which aspects of the noise prove to be of most influence. For clarity, we will refer to each step by its corresponding number as defined in the list above. Some steps are mutually exclusive (i.e. 2 and 5), so we only use the following possible combinations: 128, 158, 13568, 135678, 1237, 1236, 156, 1567, 123, and 134.

It should be noted that these pre-processed texts are not suitable for all classification methods (further discussed in Sect. 4.5). Some only require lower-casing, while others require no pre-processing at all.

[5] https://www.nltk.org/api/nltk.stem.html.

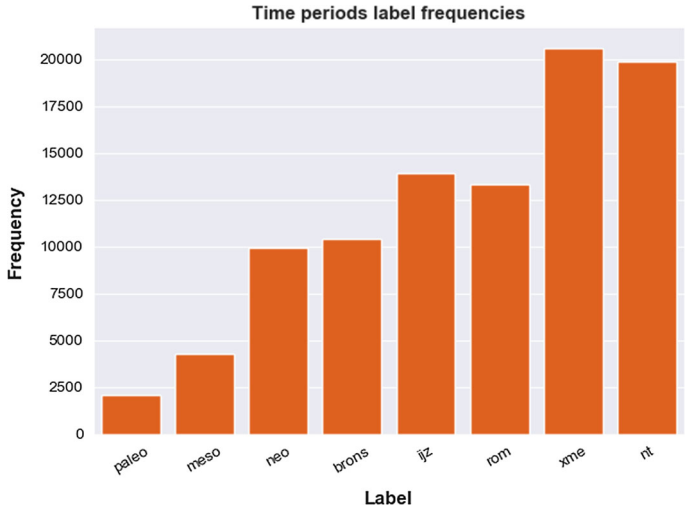

Fig. 2 An overview of the frequencies of the eight time period categories. X axis labels as per Table 2a

4.2 Document filtering

We remove all documents that have fewer than 1000 utf-8 characters. Files shorter than 1000 characters rarely contain proper text, but are appendices with only numbers, or OCRed maps resulting in a file with nonsensical characters.

In addition, we remove non-relevant documents from the data set. This relevance is based on certain terms occurring in the title, indicating it is a specific type of non-relevant document. We define two lists, the first consists of a few general terms: *notulen* (minutes), *bijlage* (appendix) and *meta* (metadata). The second list is more extensive, and includes several types of reports (RAP), working methods (PVA), requirements definitions (PVE), referential research IDs (OMN) and the aforementioned general terms. A complete overview can be found in Appendix B. For upcoming experiments, we refer to the first list consisting of general terms as *genList*, and the extensive list as *totList*.

It should be noted that while these documents are removed from our training and test set, this should not affect the usefulness of the methods on new data. Short documents that do contain useful information can still be labelled by the classifier. The document types in the genList and totList that we here exclude are most often grouped in a DANS data set with associated ID, together with the main report. When this main report has been classified, we can propagate the labels to all documents in that data set, ensuring useful metadata for all related files.

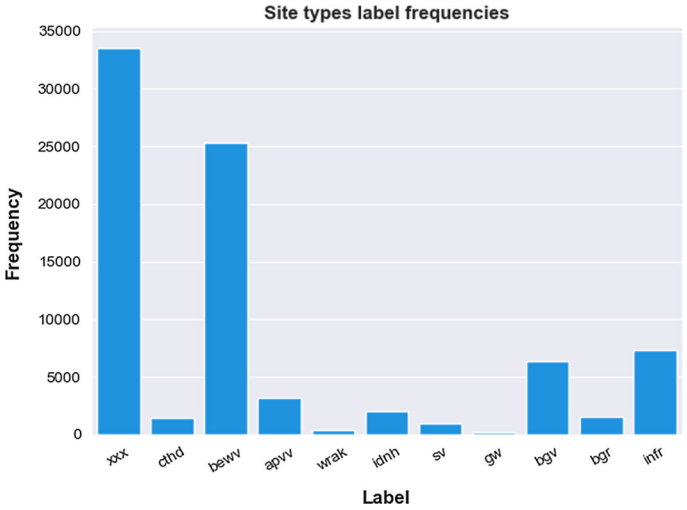

Fig. 3 An overview of the frequencies of the eleven site type categories. X axis labels as per Table 2b

4.3 Balancing the training set

As can be seen in Figs. 2 and 3, the distribution of the labels among categories is rather skewed. Some categories are not represented very well, leading to an imbalanced data set. As this might induce bias to some classifier types, we introduce two methods that may negate this. The first is balancing of the training set through under-sampling, i.e., reducing the number of documents of a class until it equals that of the class with the lowest representation. Under-sampling has been proven to be a reliable method for addressing the imbalance of a dataset regarding the distribution of present labels (Branco et al., 2015; Mohammed et al., 2020).

Another option, which primarily aims to create more valid training samples, is increasing the representation of all labels through augmentation. Here, we enlarge the training set by including files multiple times, but applying a synonym mapping function to the duplicate files to avoid bias on certain terms, while still maintaining context as much as possible. We adapt the Easy Data Augmentation (EDA) method proposed by Wei and Zou (2019). Synonyms are chosen at random with the use of the Open Dutch WordNet (Postma et al., 2016) synonym thesaurus. The augmentation should be applied to the complete corpus in order to introduce a large variety of terms, rather than merely the archaeological tokens captured within the texts. We therefore decided to make use of a thesaurus that meets this requirement, not limiting ourselves to a domain specific, in this case an archaeological, thesaurus. Contrary to the EDA method, we insert synonyms for all words longer than five characters—as opposed to a specific number of tokens based on sentence length. This is because the sentence length is in many cases simply impossible to properly determine due to noise in the text. This could potentially lead to too much semantic change in the text for it to be useful, but we

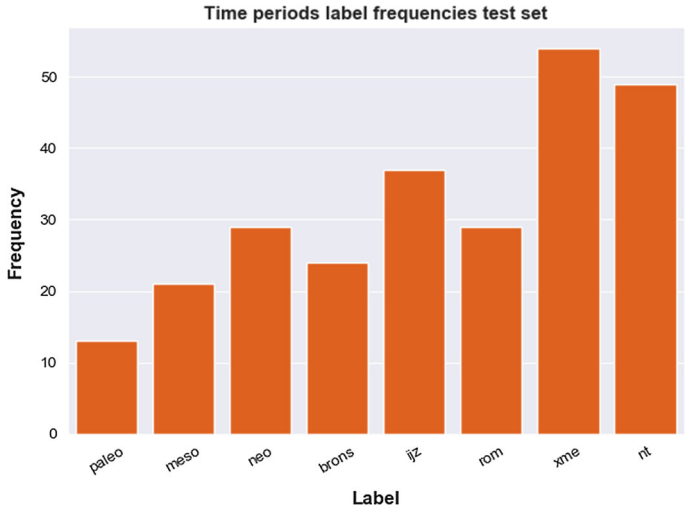

Fig. 4 An overview of the frequencies of the eight categories for time period classification, as captured within our reference set

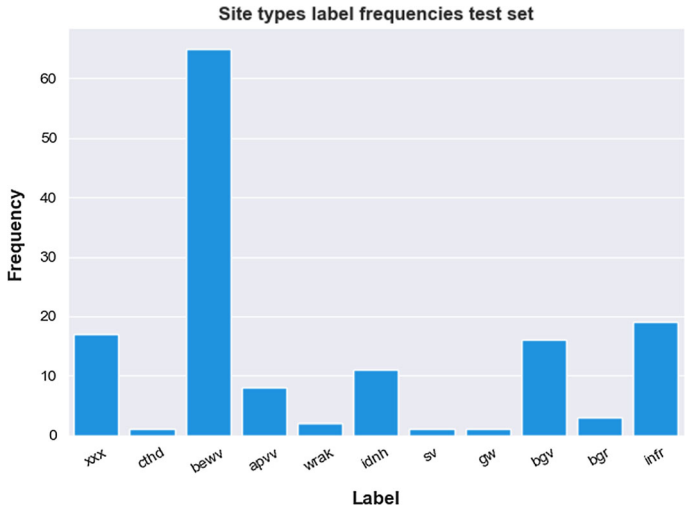

Fig. 5 An overview of the frequencies of the eleven categories for site type classification, as captured within our reference set

found this process can lead to higher performance in some cases (as further described in the Sect. 5).

4.4 Construction of a manually labelled reference set

Because of how we constructed the labels from the data, it would be impossible to ensure that all files within a randomly sampled sub-set hold only correct labels. This means that even our test set would include an unknown percentage of incorrectly labelled documents. Naturally, this is undesirable, as no valid conclusions can be drawn from a flawed test set.

In order to deal with this issue, we created a manually labelled reference test set (Brandsen et al., 2020), of which we are certain that it consists of correctly labelled documents only. As manual labelling is very time consuming, this test set consists of 'only' 100 files. Figures 4 and 5 show the frequencies for each of the categories captured within the classification of time periods and site types, respectively. The average number of labels per document is 13.9 for time periods (median: 11, max: 53) and 2.79 for site types (median: 2, max: 13).

The distributions of the test set are similar to those of the training set, as were shown in Figs. 2 and 3. The only exception is the category of label *xxx* (unknown) for the site types. This is because all files in our test set are labelled with at least one time period, and many files labelled by *xxx* (i.e., reports about sites with no finds) are not assigned any time period. A complete overview that includes all the frequencies of all main and sub-categories can be found in Appendix C.

4.5 Classification methods

We compare three methods for the classification of time periods and site types: a (naive) baseline, binary relevance, and direct multi-labelling. All methods will be trained and optimised using a train and development set, and finally evaluated on the held-out test set consisting of the manually labelled reference set mentioned above.

4.5.1 Baseline

For the baseline, we introduce the rather intuitive method of merely checking whether the label or its corresponding textual version is present within the text, and assign labels accordingly. The minimum occurrence for such tokens in the text is set to two, as often lists of ABR codes are present as period or site type lists. Naturally, these are uninformative to our research.

4.5.2 Binary relevance

We translate the multi-label task to a series of binary classification tasks, one for each category, and train a Linear Support Vector Machine (SVM) classifier for each category. We compare four feature extraction methods:

- A bag-of-words model with TF-IDF weighting;
- A Doc2Vec model (de Romas, 2019) for each individual binary classification task. The model has a vector size of 100, a window of 5, an initial learning rate

of 0.025, a minimum learning rate of 2.5e−3, and a minimum count of 5 (ignores all tokens with a frequency lower than 5). We let the model train for 5 epochs.
- Using entities as features. Besides applying pre-processing methods, we also investigate the effects when classification is done solely on extracted named entities, again using a bag-of-words model with TF-IDF weighting. We extracted entities based on the ABR ontology. Here, we extract all terms (time periods, site types, corresponding abbreviations, etc.) contained in the ontology from the text, and use this as our input.
- Same as above, but we perform the extraction of entities by means of spaCy (Honnibal & Montani, 2017), using its pre-trained Dutch model[6]. Here, we select entities from any of the following types[7]: *FAC* (groupings), *NORP* (structures) and *DATE* (dates or periods).

For the third method, we are aware that the problems with the ABR ontology as described in Sect. 3.2 will cause noise to some extent. Specifically, as no synonyms are available in the ontology, and we do not use lemmatisation or stemming, extracting terms from the text is going to have a low recall. Also, only time period names are included in the ABR, so actual dates (e.g. '1000 BCE') will not be extracted. Despite these issues, we still considered this worthwhile to experiment with, as this method can be improved by using more advanced NER methods if promising results are achieved.

4.5.3 Direct multi-labelling

Finally, we make use of BERT, a state-of-the-art classification model. We use the Simple Transformers library [8] for faster training and evaluation. Using the pre-trained bert-base-multilingual-cased model (Devlin et al., 2018), we use the following default parameter settings to evaluate the method: a train batch size of 4, gradient accumulation steps of 1, a learning rate of 3e−5, and a max sequence length of 256 due to memory constraints. The model will be trained over 3 epochs.

Initially, we limit the classification task to only the top level categories, and will use the results to determine which setting works best for any particular category.

4.6 Selection round

In summary, we have six approaches (baseline, four binary, one direct multiclass classification), ten pre-processing combinations, the option of augmenting as well as balancing the training set, and filtering files based on document title. Exploring all applicable different settings on each of these approaches will most likely lead to an abundance of scores that are far from optimal, and not very interesting. We therefore first run each of the approaches on the raw (no pre-processed versions) of

[6] https://spacy.io/models/nl.

[7] https://spacy.io/api/annotation#named-entities.

[8] https://pypi.org/project/simpletransformers/.

the documents, and determine how each method performs with respect to the baseline and one another. To limit the aforementioned parameter exploration, we will continue with the two best performing approaches for the time periods and site types, based on the F1 score.

One aspect that should be taken into account is that BERT in particular should theoretically already be performing closer to optimal compared to the binary translation approaches, as pre-processing is not required for this method.

5 Results

In this section, we present our results. We first determine how each approach performs on the data set with no modifications, and then select the top two performing approaches for further research. We then investigate the effects of different parameter settings, determine the best possible method per category, and finally perform the classification task on all categories.

5.1 Selection round

We have a baseline and five approaches we will evaluate first. The obtained precision, recall and F1 scores can be seen in Table 4. All scores are the macro average over all categories within the corresponding field. For TF-IDF, D2V, ONT and SCY (acronyms explained in the table caption), a linear support vector classifier was used. For BERT, we used the pre-trained bert-base-multilingual-cased model[9]. The two best performing approaches are highlighted in green.

For the time periods, the baseline F1 score of 0.358 is substantially outperformed by the other five approaches. Even without pre-processing, the four binary classification approaches, TF-IDF, D2V, ONT and SCY already lead to decent results. As highlighted, TF-IDF and ONT score the highest, the former by a noticeable amount. BERT unfortunately does not yield very promising results, particularly so as this approach does not require any prior pre-processing on the texts.

For the site types, we find that the baseline performs better than both SCY and BERT, the latter two yielding an F1 score of less than 0.15. Again, TF-IDF and ONT give the best results, though only by a very small, almost negligible margin when comparing ONT to D2V. Nevertheless, we continue with TF-IDF and ONT for both time periods and site types, and will now look at pre-processing optimisation.

5.2 Pre-processing optimisation

We applied a brute force approach, trying out all 176 combinations of pre-processing steps, balancing/augmenting the training set, and further pruning the training set based on document titles.

[9] https://huggingface.co/transformers/pretrained_models.html.

Table 4 Overview of the scores for each method

Performance metrics time periods				Performance metrics site types			
Approach	**Precision**	**Recall**	**F1**	**Approach**	**Precision**	**Recall**	**F1**
Baseline	0.500	0.318	0.358	Baseline	0.161	0.622	0.232
TF-IDF	0.848	0.621	0.703	TF-IDF	0.633	0.355	0.408
D2V	0.747	0.500	0.577	D2V	0.313	0.282	0.254
ONT	0.854	0.506	0.602	ONT	0.434	0.270	0.259
SCY	0.795	0.484	0.565	SCY	0.272	0.140	0.121
BERT	0.745	0.519	0.585	BERT	0.225	0.151	0.146

Abbreviations refer to the following: *TF-IDF* Sklearn, linear SVM with TF-IDF weights, *D2V* Sklearn, linear SVM with Doc2Vec vectors, *ONT* Sklearn, linear SVM classification based on ontology extracted entities, *SCY* Sklearn, linear SVM classification based on spaCy retrieved entities

Table 5 Overview of the top ten F1 scores for time period classification

Dev rank	Test rank	PP	Aug	Precision	Recall	F1
1	3	1237	0	0.873	0.639	0.719
2	8	134	0	0.856	0.602	0.681
3	6	134	2	0.869	0.597	0.692
4	7	123	2	0.865	0.602	0.684
5	5	158	0	0.857	0.635	0.709
6	1	128	2	0.873	**0.674**	**0.752**
7	4	123	0	**0.880**	0.631	0.711
8	2	128	0	0.879	0.652	0.730
9	10	1237	2	0.874	0.568	0.658
10	9	1236	0	0.863	0.605	0.680

Highest scores highlighted in bold

PP = numerical values referring to pre-processing steps as described in Sect. 4.1, Aug = number of augments of the training set

The performance metrics were determined by averaging the F1 scores over three separate evaluation rounds. During each round, the training set was split into a 4:1 ratio, retaining a suitable training set size and introducing a smaller development set.

Tables 5 and 6 show the top ten performing settings, ordered by obtained F1 scores on the development set, but showing the performance metrics on the held out test set. The second column, labelled *Test Rank* indicates which ranking the top ten performance settings on the development set achieve when these same settings are applied to the test set. The ranking captured within the *Test Rank* column thus reflects the ordering of the *F1 scores*, which are shown in the rightmost column. The top ten combinations all use the bag-of-words model with TF-IDF weighting, classifier Linear SVC, no balancing and the GenList document pruning list, so these are not mentioned in the tables.

The results show that rather short combinations consisting of only three or four pre-processing steps lead to the overall highest results in combination with the SVM classifier. Steps 1, 2 and 3 occur almost everywhere. These are lowercasing, removing punctuation marks and removing abundant white space, which are expected to help with classification as these are commonly used.

Table 6 Overview of the top ten F1 scores for site types classification

Dev rank	Test rank	PP	Aug	Precision	Recall	F1
1	7	123	2	0.626	0.360	0.410
2	4	13,568	2	0.637	0.464	0.496
3	3	128	0	0.601	0.462	0.498
4	9	1236	0	0.542	0.347	0.379
5	10	134	2	0.539	0.330	0.366
6	1	158	2	0.640	**0.499**	**0.542**
7	2	128	2	0.702	0.469	0.510
8	8	123	0	0.538	0.345	0.390
9	6	1237	2	**0.715**	0.442	0.482
10	5	1237	0	0.609	0.447	0.484

Highest scores highlighted in bold

PP = numerical values referring to pre-processing steps as described in Sect. 4.1, Aug = number of augments of the training set

Augmentation of the training set does not necessarily seem to have a positive effect on the classification process as it only leads to higher F1 scores with certain pre-processing combinations. Finally, we can make the observation that filtering files based on terms included in genList also leads to better performance for both time periods and site types, whereas totList does not appear in any of the top ten rankings.

Despite these scores being the average over three runs, the balancing and augmentation is a rather randomised process. It is therefore possible that a lot of 'bad' or 'good' files are filtered out, i.e., files that have (un)informative content. This would mean that the performance metrics could vary slightly when the experiments are to be repeated, perhaps resulting in a different ranking.

Lastly, the development and test ranking orders provide some interesting insight into how representative the defined development sets were compared to the reference set. We can see that for both time periods and site types, the best performing settings on the test set are found at rank six for the development set. As the optimal development and test F1 scores differ quite heavily from one another, the quality of the development sets do not match that of the test set. This was to be expected, as the training set, and therefore the development sets contain an unknown percentage of wrong labels.

5.3 Best methods per category

The above section shows which approach and parameter settings lead to the highest average F1 scores, and here we investigate if we can achieve a higher average F1 score by combining the best approaches and settings for each individual category. The results for time periods and site types are shown in Tables 7 and 8, respectively.

For time periods, combining the best method per individual category leads to an average F1 score of 0.710, which is a slight decrease compared to the 0.719 of the settings with the best F1 average over all categories. This again can be explained by the quality of the development sets: by using the optimal parameter settings for a category obtained on the development set, it unfortunately does not imply that these

Table 7 Overview of the best methods per individual category for time period classification and the overall average of these best methods

Category	PP	Aug	Bal	List	Precision	Recall	F1 score
paleo	123	2	No	Gen	1.0	0.385	0.555
meso	134	2	No	Gen	1.0	0.550	0.710
neo	123	2	Yes	Gen	0.653	0.630	0.642
brons	158	2	No	Gen	0.714	0.435	0.541
ijz	134	0	Yes	Gen	0.828	0.828	0.828
rom	128	0	No	Gen	0.952	0.741	0.833
xme	1236	0	No	Gen	0.764	0.823	0.792
nt	134	0	No	Gen	0.722	0.848	0.780
Average	–	–	–	–	0.829	0.655	0.710

Column names yield the meaning as provided in the previous section

Table 8 Overview of the best methods per individual category for site type classification and the overall average of these best methods

Category	PP	Aug	Bal	List	Precision	Recall	F1 score
xxx	1237	0	No	Tot	0.342	0.765	0.473
cthd	156	2	No	Gen	1.0	1.0	1.0
bewv	123	0	No	Gen	0.810	0.557	0.660
apvv	128	0	No	Gen	0.667	0.286	0.400
wrak	13568	0	No	Tot	1.0	0.500	0.667
idnh	123	2	No	Gen	0.800	0.444	0.571
sv	1236	2	No	Gen	1.0	1.0	1.0
gw	134	2	No	Gen	0.0	0.0	0.0
bgv	156	2	No	Gen	0.875	0.538	0.667
bgr	128	2	No	Gen	0.0	0.0	0.0
infr	1237	2	No	Gen	0.875	0.389	0.538
Average	–	–	–	–	0.669	0.498	0.543

Column names yield the meaning as provided in the previous section

settings are (close to) optimal on the test set. This phenomenon is similar to that observed in the previous section, were the best parameter settings for the test set ranked sixth on the development set. For the site types, the opposite shows, as we find an average increase of 0.133 compared to the highest scoring settings on the development set. Moreover, the F1 score of 0.542—the result of optimal settings for the test set—is met. It has to be noted that we find F1 scores of 0.0. These categories are barely represented within our test set, and for these it is difficult to determine the quality of the classification process: a recall of 0.0 is frequent.

Table 9 An overview of the F1 scores for all main and sub-categories for time period classification

All time periods categories: obtained F1 scores overview									
Label	F1	Label	F1	Label	F1	Label	F1	Label	F1
paleo	0.555	neov	0.591	bronsm	0.583	romvb	0.700	vmec	0.439
paleov	0.600	neova	0.605	bronsma	0.522	romm	0.833	vmed	0.439
paleom	0.667	neovb	0.667	bronsmb	0.640	romma	0.809	lme	0.800
paleol	0.500	neom	0.619	bronsl	0.500	rommb	0.833	lmea	0.756
paleola	0.500	neoma	0.537	**ijz**	0.828	roml	0.780	lmeb	0.787
paleolb	0.500	neomb	0.585	ijzv	0.750	romla	0.800	**nt**	0.780
meso	0.710	neol	0.681	ijzm	0.644	romlb	0.810	nta	0.738
mesov	0.455	neola	0.667	ijzl	0.719	**xme**	0.792	ntb	0.764
mesom	0.500	neolb	0.696	**rom**	0.833	vme	0.455	ntc	0.689
mesol	0.571	**brons**	0.541	romv	0.700	vmea	0.450		
neo	0.742	bronsv	0.483	romva	0.700	vmeb	0.450		

Main categories are shown in bold

Table 10 An overview of the F1 scores for the main and sub-categories for site type classification

All site type categories: obtained F1 scores overview									
Label	F1	Label	F1	Label	F1	Label	F1	Label	F1
cthd	1.0	bewv.hp	0.000	idnh.hkb	0.0	bgv.x	0.0	**bgr**	0.0
cthd.klo	1.0	bewv.bext	0.667	idnh.ll	0.0	bgv.gvc	0.667	bgr.gvic	0.0
bewv	0.857	**apvv**	0.400	idnh.m	0.0	bgv.gvi	0.0	**infr**	0.571
bewv.x	0.756	apvv.x	0.0	idnh.pb	0.0	bgv.gvx	0.500	infr.x	0.0
bewv.vx	0.0	apvv.cf	0.0	idnh.vb	0.0	bgv.kh	0.0	infr.weg	0.0
bewv.vlp	0.0	apvv.la	0.333	idnh.mb	0.0	bgv.ghv	0.500	infr.per	0.800
bewv.kwb	0.0	**wrak**	0.667	**sv**	1.0	bgv.cjbp	0.0	infr.kan	0.667
bewv.ht	0.0	wrak.schip	0.667	sv.x	1.0	bgv.uv	0.400	infr.brug	0.667
bewv.vic	0.0	**idnh**	0.800	**gw**	0.0	bgv.gx	0.0	infr.dij	0.889
bewv.sk	0.667	idnh.x	0.286	gw.vw	0.0	bgv.vg	0.0	**xxx**	0.473
bewv.rv	1.0	idnh.tn	0.0	**bgv**	0.667	bgv.dier	0.0		

Sub-categories not present within the reference test set are not included. Again, main categories are shown in bold

The MultNB classifier does not appear in the top ten. We expected to see that balancing the training set would have a positive effect on the classification process for this classifier, but this is not reflected by our results. However, it is interesting to see that balancing the training set has a positive effect on the classification process

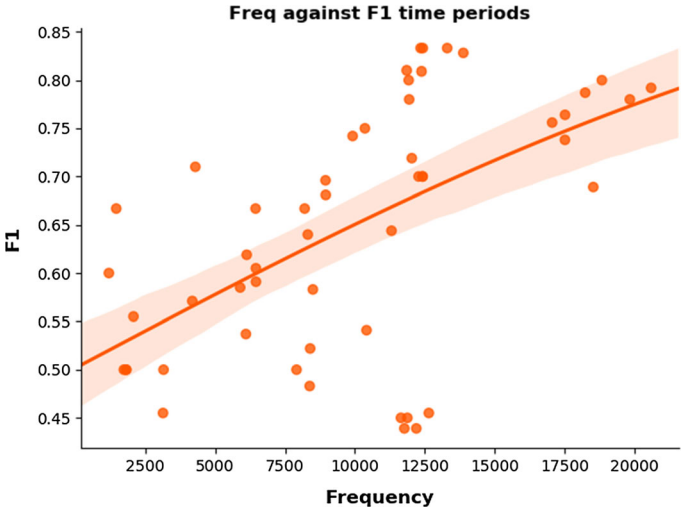

Fig. 6 Plot of the frequency of time period labels and the associated F1 score for that label. A trend line has been added to illustrate the correlation (Pearson's $r = 0.56$)

of SVM for *neo* and *ijz*, despite the theoretical unbalanced data set 'protection'. Again, this can be explained by the random influence of the balancing and augmenting process, as 'bad' files get filtered out.

We have determined which settings work best for each main category, the next step is to perform the classification task on all sub-categories by using the settings per corresponding main category. As not all sub-categories for site types are present within our test set, we will only focus on those that were. The full classification results can be seen in Tables 9 and 10.

For any set of sub-categories, we expected to find a lower average F1 score than the corresponding main category, as there are most likely less distinctive terms between sub-categories. This indeed seems to be case for the majority of the categories, but a few exceptions for both time periods and site types are present. We note that in some cases for the site types, F1 scores of 1.0 are found. These (sub-)categories are only represented once. Nevertheless, it does imply that the classifier returns a perfect prediction on our test set. We also find numerous F1 scores of 0.0, which as mentioned earlier is the result of frequent recall values of 0.0.

Such scores are not very indicative of the quality of the classification process itself, but rather implies an insufficient amount of labelled data for that category. For completeness however, we decided not to omit these results from aforementioned tables.

To further illustrate the relation between the frequency of a label in the training set and the achieved F1 scores, we plotted these in Figs. 6 and 7. We can see that—as expected—the higher the frequency of the label is, the higher the performance, as illustrated by the trend lines. We also note that the trend lines are not flattening out, which indicates that adding more training data might be beneficial for all categories, not just the less frequent ones.

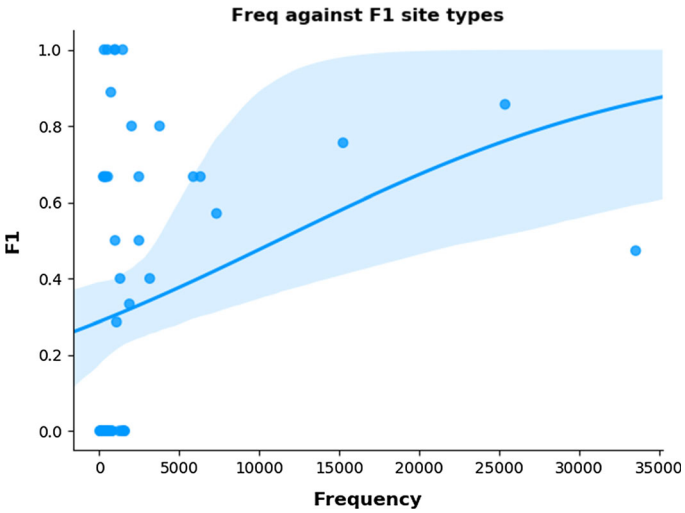

Fig. 7 Plot of the frequency of subject labels and the associated F1 score for that label. A trend line has been added to illustrate the correlation (Pearson's $r = 0.28$)

6 Conclusion

In this paper, we have described our approach for the multi-labelling of Dutch archaeological excavation reports for time periods and site types. In this section we answer our research questions and propose future work.

> *Which combination(s) of text pre-processing steps, data augmentation/ balancing, document pre-selection and classification method yields the highest F1 scores?*

We tested many combinations of pre-processing steps, and found that lowercasing, removing punctuation marks and trimming white space are most valuable on average, which is expected as these steps are used widely in text classification problems. Balancing the data set did not lead to better results, and augmentation helped in only some cases, so we can not draw any conclusions on this. Pruning the data set by using the standard filename list proved to be most effective. As for the classification method, using a linear SVM proved to be optimal. In addition, we found that classification on extracted entities by means of the ontology did not yield very promising results.

> *Are the best combinations the same across the different categories and labels, or do specialised combinations per category yield better results?*

We investigated whether optimising the methods per (sub-)category leads to higher performance. We found that the optimal parameter settings per individual category for the time periods actually lead to a lower averaged F1 score when compared to the top performing setting over all categories at once. For site types the F1 score is the same. It suggests that for these kinds of classification problems, using the same

parameters for all the categories is not only better, but also much simpler as only one model needs to be trained, instead of a model for each category.

To what extent can we classify excavation reports into time periods and site types?

Our overall aim was to test how well we could classify excavation reports, and we found that despite the frequent low quality of both texts and labels, our classification models lead to decent quality when compared to similar studies. For the classification in eight time periods, we obtained an F1 score of 0.752 with settings that were found to be optimal on the held-out test set. These included only a few pre-processing steps, no balancing, and a small selection for filtering documents based on their titles. For the classification in eleven site type categories, we obtained an F1 score of 0.542 with highly similar settings, except for a single different text pre-processing step (removal of non-alphabetical marks instead of removal of punctuation marks) and the augmentation of the training set.

One caveat to these results is that there is a large deviation in the results obtained with different partitions of the data, with the top ten highest scoring partitions of the development set leading to F1 scores on the test set ranging from 0.68 to 0.75 for time period classification and from 0.36 to 0.54 for site type classification.

We expected to see that the average F1 scores over a set of sub-categories would be lower than that of the corresponding main category. This was indeed the case apart from a few exceptions. We argued that this phenomenon is caused by a smaller number of distinctive terms for sub-categories when compared to solely main categories.

As predicted, the limited input sequence of 256 for BERT led to quite disappointing results, considering this method is regarded as a state-of-the-art approach for multi-label classification tasks. In particular for the site type classification, performance metric scores for BERT were almost bottom tier.

6.1 Future work

There are several aspects that could prove interesting for follow-up research. At the moment, we are dealing with a data set that has manually assigned metadata for the entire collection. This means our methods are not tested on unlabelled, or partially labelled data. It would be interesting to research this, to see to what extent the usefulness of the metadata increases. We plan to do this research when we receive reports without metadata in a follow-up project.

As we were particularly concerned about the effect the quality of the labels and the texts would have on the classification process, we put more emphasis on the application of exploratory parameter settings based on statistics on observations, rather than using all the five approaches. It could prove to be interesting to apply the parameter settings to each of these, and eventually perform hyper-parameter optimisation. Ideally, we would like to create a manually labelled training set to increase the quality of the data, and determine how this affects the performance of our methods. Due to time constraints we have not been able to do so in yet. If this

proves too time-consuming, an alternative might be k-fold validation to average out the difference in label quality across the training set.

Initially, we opted for NER based classification by means of a specifically designed NER tool for archaeological named entities. Unfortunately, this tool had not been fully developed yet, and could not be used. SpaCy based NER classification already lead to promising results—scoring second highest for both time periods and site types—despite a lack of entity categories that were specific to our type of documents. If such categories were to be extracted however, classification on such entities might lead to even better results.

A third aspect that could be addressed is that of balancing: we might be able to determine which files are often included in a training set that leads to lower performance. This would arguably imply that such files are either uninformative, or have erroneous labels. Removing these documents will most likely lead to higher overall performance.

Furthermore, there is the option of optimising the BERT approach. Currently we only use the first 256 tokens of a text due to memory and framework constraints. Distinctive and characteristic terms for categories could therefore be missing in data used for either training or eventual classification, leading to lower performance. Increasing the token limit, or potentially classifying smaller segments, might give us better results.

Finally, an expansion of the test set could be introduced in order to enhance the representation of the categories. This in particular applies to the categories of site types. As discussed in Sect. 5.3, we find an F1 score for numerous site type categories to be equal to 0.0 or 1.0. Because of the low representation of these categories, such scores are not meaningful, and therefore do not properly reflect on the quality of the classification process.

Author contributions Both authors contributed to the study conception, design and paper writing. Material preparation, data collection and supervision were performed by Alex Brandsen. Data pre-processing, classification pipelines and analysis were performed by Martin Koole. The manuscript was prepared by both authors.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendices

A: Category frequencies

Table 11 An overview of the frequencies for all site type categories

Site type categories frequency overview									
Label	Freq	Label	Freq	Label	Freq	Label	Freq	Label	Freq
xxx	33,532								
cthd	1463	bewv.wp	10	**idnh**	2015	sv.vorg	0	bgv.meg	9
cthd.x	379	bewv.n	0	idnh.x	1074	sv.bsb	0	**bgr**	1502
cthd.klo	299	bewv.rv	501	idnh.tk	0	**gw**	169	bgr.gvic	1502
cthd.kpl	24	bewv.stel	4	idnh.tn	128	gw.x	40	**infr**	7327
cthd.sgmw	0	bewv.bw	0	idnh.br	36	gw.vw	58	infr.x	1248
cthd.kerk	581	bewv.hp	1566	idnh.zp	0	gw.hout	0	infr.weg	1575
cthd.rcp	372	bewv.th	0	idnh.sb	71	gw.ijw	9	infr.dam	53
cthd.oloc	1	bewv.inka	0	idnh.hkb	127	gw.zw	3	infr.werf	0
cthd.temp	2	bewv.sv	0	idnh.bb	5	gw.kw	50	infr.gem	5
bewv	25,264	bewv.bext	5872	idnh.ll	112	gw.griw	0	infr.rede	0
bewv.x	15,236	bewv.vkm	63	idnh.hb	4	gw.mw	4	infr.per	3766
bewv.lg	89	bewv.tw	388	idnh.m	150	gw.vsw	8	infr.strek	0
bewv.wb	51	bewv.lw	130	idnh.rom	1	**bgv**	6317	infr.wat	228
bewv.sch	65	**apvv**	3152	idnh.wam	2	bgv.x	731	infr.dui	221
bewv.vx	815	apvv.x	1415	idnh.wim	1	bgv.gvc	522	infr.vijv	0
bewv.vlp	102	apvv.vw	0	idnh.gp	0	bgv.tpgb	0	infr.kan	273
bewv.lk	0	apvv.vk	166	idnh.pb	227	bgv.gvi	536	infr.slu	103
bewv.ct	0	apvv.vs	6	idnh.vb	312	bgv.gvx	983	infr.kslu	0
bewv.cstl	5	apvv.stel	0	idnh.mb	388	bgv.kh	605	infr.lv	0
bewv.mbh	125	apvv.ek	0	idnh.mbnf	0	bgv.rgv	1	infr.hav	982
bewvv.pls	0	apvv.cf	23	idnh.mbf	0	bgv.ghv	2476	infr.kade	1
bewv.kwb	490	apvv.dp	142	idnh.kb	2	bgv.bhv	0	infr.vweg	7
bewv.ht	332	apvv.la	1879	**sv**	971	bgv.vgv	0	infr.brug	256
bewv.aw	7	apvv.ak	0	sv.x	971	bgv.cjbp	536	infr.dok	0
bewv.dump	0	apvv.tuin	20	sv.obsb	0	bgv.uv	1295	infr.vs	0
bewv.vic	332	apvv.pdek	2	sv.ijz	0	bgv.gh	1221	infr.vrde	1
bewv.kaze	0	**wrak**	384	sv.h	0	bgv.gx	731	infr.spre	0
bewv.fort	8	wrak.schip	384	sv.lad	0	bgv.vg	163	infr.watw	11
bewv.sk	2470	wrak.vlgtg	5	sv.hijz	0	bgv.dier	131	infr.dij	721

Main categories are denoted in bold

Table 12 An overview of the frequencies for all time period categories

Time periods categories frequency overview									
Label	Freq	Label	Freq	Label	Freq	Label	Freq	Label	Freq
paleo	2077	neov	6459	bronsm	8494	romvb	12414	vmec	11767
paleov	1197	neova	6456	bronsma	8397	romm	12427	vmed	12194
paleom	1460	neovb	6445	bronsmb	8312	romma	12381	lme	18832
paleol	1816	neom	6127	bronsl	7910	rommb	12348	lmea	17053
paleola	1732	neoma	6098	**ijz**	13876	roml	11939	lmeb	18235
paleolb	1816	neomb	5893	ijzv	10356	romla	11921	**nt**	19833
meso	4290	neol	8954	ijzm	11307	romlb	11850	nta	17511
mesov	3133	neola	8200	ijzl	12033	**xme**	20593	ntb	17514
mesom	3152	neolb	8947	**rom**	13299	vme	12642	ntc	18525
mesol	4180	**brons**	10414	romv	12421	vmea	11645		
neo	9916	bronsv	8380	romva	12275	vmeb	11874		

Main categories are denoted in bold

B: Filter list

Table 13 An overview of different types of lists and included terms

Terms used for document filtering	
List name	Terms
genList	notulen, bijlage, meta
rapList	dagrapport, dag_rapport, weekrapport, week_rapport, weekverslag, week_verslag, logboek
pvaList	draaiboek, plan_van_aanpak, pva
omnList	onderzoeksmeldingsnummer, onderzoeksmeldings_nummer, onderzoeks_meldings_nummer
totList	rapList + pvaList + pveList + omnList + genList

C: Category frequencies test set

Table 14 An overview of the frequencies for all time period categories captured by the reference test set

Time periods categories frequency overview test set									
Label	Freq	Label	Freq	Label	Freq	Label	Freq	Label	Freq
paleo	13	neov	24	bronsm	15	romvb	25	vmec	30
paleov	7	neova	24	bronsma	14	romm	27	vmed	29
paleom	8	neovb	24	bronsmb	15	romma	27	lme	48
paleol	12	neom	23	bronsl	18	rommb	27	lmea	41
paleola	12	neoma	23	**ijz**	37	roml	25	lmeb	48
paleolb	12	neomb	23	ijzv	34	romla	25	**nt**	49
meso	21	neol	26	ijzm	28	romlb	25	nta	42
mesov	17	neola	25	ijzl	30	**xme**	54	ntb	43
mesom	18	neolb	26	**rom**	29	vme	30	ntc	39
mesol	20	**brons**	24	romv	25	vmea	28		
neo	29	bronsv	19	romva	25	vmeb	28		

Main categories are denoted in bold

Table 15 An overview of the F1 scores for the main and sub-categories for site type classification as captured by the reference test set

Site type categories frequency overview test set									
Label	Freq	Label	Freq	Label	Freq	Label	Freq	Label	Freq
cthd	1	bewv.hp	7	idnh.hkb	2	bgv.x	4	**bgr**	3
cthd.klo	1	bewv.bext	3	idnh.ll	1	bgv.gvc	2	bgr.gvic	3
bewv	65	**apvv**	8	idnh.m	1	bgv.gvi	3	**infr**	19
bewv.x	53	apvv.x	3	idnh.pb	1	bgv.gvx	3	infr.x	1
bewv.vx	1	apvv.cf	1	idnh.vb	2	bgv.kh	1	infr.weg	4
bewv.vlp	1	apvv.la	4	idnh.mb	2	bgv.ghv	6	infr.per	6
bewv.kwb	1	**wrak**	2	**sv**	1	bgv.cjbp	3	infr.kan	2
bewv.ht	10	wrak.schip	2	sv.x	1	bgv.uv	4	infr.brug	2
bewv.vic	1	**idnh**	11	**gw**	1	bgv.gx	4	infr.dij	5
bewv.sk	4	idnh.x	4	gw.vw	1	bgv.vg	1	**xxx**	17
bewv.rv	1	idnh.tn	1	**bgv**	16	bgv.dier	1		

Sub-categories not present within the reference test set are not included. Again, main categories are denoted in bold

References

Branco, P., Torgo, L., & Ribeiro, R. (2015). A survey of predictive modelling under imbalanced distributions.

Brandsen, A., & Koole, M. (2020). Alexbrandsen/archaeo-labelling-gold-standard: First version (Version v1.0.). *Zenodo*. https://doi.org/10.5281/zenodo.4115747

Brandsen, A., Lambers, K., Verberne, S., & Wansleeben, M. (2019). User requirement solicitation for an information retrieval system applied to Dutch grey literature in the archaeology domain. *Journal of Computer Applications in Archaeology, 2*(1), 21–30. https://doi.org/10.5334/jcaa.33

Brandsen, A., Verberne, S., Lambers, K., & Wansleeben, M. (2021). Usability evaluation for online professional search in the Dutch archaeology domain. arXiv. http://arxiv.org/abs/2103.04437

Brandsen, A., Verberne, S., Wansleeben, M., & Lambers, K. (2020). Creating a dataset for named entity recognition in the archaeology domain. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 4573–4577). https://doi.org/10.5281/zenodo.3544544

Brandt, R., Drenth, E., Montforts, M., Proos, R., Roorda, I., & Wiemer, R. (1992). Archeologisch Basisregister. Tech. rep., Rijksdienst voor Cultureel Erfgoed, Amersfoort.

Cherman, E. A., Monard, M. C., & Metz, J. (2011). Multi-label problem transformation methods: A case study. *CLEI Electronic Journal, 14,* 4–4.

de Romas, R. (2019). *Multi-label text classification for ground lease documents*. Master's thesis, University of Amsterdam.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR. http://arxiv.org/abs/1810.04805

Fiorucci, M., Khoroshiltseva, M., Pontil, M., Traviglia, A., Del Bue, A., & James, S. (2020). Machine learning for cultural heritage: A survey. *Pattern Recognition Letters, 133,* 102–108.

Glyph & Cog LLC: pdftotext. (1996). https://www.xpdfreader.com/pdftotext-man.html

Golub, K., Hagelbäck, J., & Ardö, A. (2020). Automatic classification of swedish metadata using dewey decimal classification: A comparison of approaches. *Journal of Data and Information Science, 5*(1), 18. https://doi.org/10.2478/jdis-2020-0003

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear, still as of February 2020.

Jeffrey, S., Richards, J., Ciravegna, F., Waller, S., Chapman, S., & Zhang, Z. (2009). The archaeotools project: Faceted classification and natural language processing in an archaeological context. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 367*(1897), 2507–2519. https://doi.org/10.1098/rsta.2009.0038

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98* (pp. 137–142). Springer.

Kleppe, M., Hendrickx, I., Veldhoen, S., Brandsen, A., Vos, H. D., Goes, K., Huang, L., Huurdeman, H., Kim, A., Mesbah, S., Reuver, M., Wang, S., & Zijdeman, R. (2019). (Semi-) automatic cataloguing of textual cultural heritage objects. Tech. rep. KB (National Library of the Netherlands), Den Haag.

Laza, R., Pavón, R., Reboiro-Jato, M., & Fdez-Riverola, F. (2011). Evaluating the effect of unbalanced data in biomedical document classification. *Journal of Integrative Bioinformatics, 8*(3), 177. https://doi.org/10.1515/jib-2011-177

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In *2020 11th International Conference on Information and Communication Systems (ICICS)* (pp. 243–248). https://doi.org/10.1109/ICICS49469.2020.239556

Paijmans, H., & Brandsen, A. (2010). Searching in archaeological texts: Problems and solutions using an artificial intelligence approach. *PalArch's Journal of Vertebrate Palaeontology, 7*(2), 1–6.

Postma, M., van Miltenburg, E., Segers, R., Schoen, A., & Vossen, P. (2016). Open Dutch WordNet. In *Proceedings of the Eight Global Wordnet Conference* (pp. 300–308). Bucharest, Romania.

Powers, D. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology, 2*(1), 37–63.

Renfrew, C., & Bahn, P.G. (2019). *Archaeology: Theories, methods and practice* (8th ed.). Thames and Hudson.

Sasaki, Y. (2007). The truth of the F-measure. Tech. rep., School of Computer Science, University of Manchester, Manchester. https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf

Sporleder, C. (2010). Natural language processing for cultural heritage domains. *Language and Linguistics Compass, 4*(9), 750–768. https://doi.org/10.1111/j.1749-818X.2010.00230.x

Vlachidis, A., & Tudhope, D. (2012). A pilot investigation of information extraction in the semantic annotation of archaeological reports. *International Journal of Metadata, Semantics and Ontologies, 7,* 222–235. https://doi.org/10.1504/IJMSO.2012.050183

Vlachidis, A., & Tudhope, D. (2016). A knowledge-based approach to information extraction for semantic interoperability in the archaeology domain. *Journal of the Association for Information Science and Technology, 67*(5), 1138–1152. https://doi.org/10.1002/asi.23485

Vlachidis, A., Tudhope, D., Wansleeben, M., Azzopardi, J., Green, K., Xia, L., & Wright, H. (2017). D16.4: Final report on natural language processing. Tech. rep., ARIADNE. http://legacy.ariadne-infrastructure.eu/wp-content/uploads/2019/01/D16.4_Final_Report_on_Natural_Language_Processing_Final.pdf

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. CoRR. http://arxiv.org/abs/1901.11196