

# A New Benchmark for NLP in Social Sciences: Evaluating the Usefulness of Pre-trained Language Models for Classifying Open-ended Survey Responses

Maximilian Meidinger and Matthias Aßenmacher<sup>a</sup>

Department of Statistics, Ludwig-Maximilians-Universität, Munich, Germany

**Keywords:** Benchmark, Multi-label Classification, Open-ended Responses, Transfer Learning, Pre-trained Language Models.


**Abstract:** In order to evaluate transfer learning models for Natural Language Processing on a common ground, numerous general domain (sets of) benchmark data sets have been established throughout the last couple of years. Primarily, the proposed tasks are classification (binary, multi-class), regression or language generation. However, no benchmark data set for (*extreme*) *multi-label* classification relying on full-text inputs has been proposed in the area of social science survey research to this date. This constitutes an important gap, as a common data set for algorithm development in this field could lead to more reproducible, sustainable research. Thus, we provide a transparent and fully reproducible preparation of the 2008 American National Election Study (ANES) data set, which can be used for benchmark comparisons of different NLP models on the task of multi-label classification. In contrast to other data sets, our data set comprises full-text inputs instead of bag-of-words representations or similar. Furthermore, we provide baseline performances of simple logistic regression models as well as performance values for recently established transfer learning architectures, namely BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019).

## 1 INTRODUCTION

The quasi-standard method in machine learning to determine the performance of a newly proposed method is to evaluate it on benchmark data sets. The same applies for the evaluation of pre-trained language models frequently utilized for transfer learning in Natural Language Processing (NLP). Collections of benchmark data sets for different natural language understanding (NLU) tasks (Rajpurkar et al., 2016; Lai et al., 2017; Wang et al., 2018) have gained massive popularity among researchers in this field. These benchmark collections stand out mainly due to two aspects: They are extremely well documented with respect to their creation and they are fixed with respect to the train-test split and the applied evaluation metrics. Furthermore they provide public leaderboards<sup>1</sup>, where the results of submitted models are displayed in a unified fashion. For the majority of the proposed benchmark data sets the task is either a binary or a multi-class classification task (cf. data sets from

Wang et al. (2018)). In the context of social science survey research, however, to our knowledge no existing (*extreme*) multi-label data sets (Lewis et al., 2004; Mencia and Fürnkranz, 2008) have been used for performance evaluation by any of the current state-of-the-art (SOTA) transfer learning models. These, and other (tabular) multi-label data sets can e.g. be found in repositories like MULAN.

In the social sciences, especially in survey research, definitive standards for raw data formatting of open-ended survey questions have not yet been established to our knowledge. This is not to say that there exist no current standards for handling and organizing survey research data in general (Inter-University Consortium For Political And Social Research (ICSPR), 2012; CESSDA Training Team, 2020) or the metadata describing the primary data (Vardigan et al., 2008; Hoyle et al., 2016). Yet, for open-ended survey questions and their *coding*<sup>2</sup>, these standards have not been well established, apart from descriptions of best practices by some authors (Züll, 2016; Lupia,

<sup>a</sup>  <https://orcid.org/0000-0003-2154-5774>

<sup>1</sup> e.g. <https://gluebenchmark.com/leaderboard>

<sup>2</sup> The process of manually assigning survey responses to pre-defined sets of labels (*codes*) is known as *coding*.

2018b,a).

Our data set preparation represents a novelty since it combines an interesting use-case (multi-label classification) for NLP models in Social Sciences with a fully reproducible pre-processing resulting in *full-text strings* as inputs. Note that this combination is not yet included in the benchmark collections mentioned above<sup>3</sup>. Thus, in the spirit of the growing overall need for standardized data sets and for reproducibility, we provide a description (cf. Sec. 2), an overview on previous use of this data set (cf. Sec. 3) and a thoroughly described pre-processing (cf. Sec. 4.1) of the ANES 2008 data, which enables its usage for benchmark comparisons for multi-label classification. Baseline performance values for a simple machine learning model as well as for more recently proposed transfer learning architectures are provided in Sec. 5.

## 2 THE "AMERICAN NATIONAL ELECTION STUDIES" SURVEY

The American Election Studies (ANES) provide high-quality data for political and social science research by conducting surveys on political participation, public opinion and voting behavior since 1948. To fulfill this commitment, ANES conducts a series of biennial election studies which cover these topics, sometimes extended by surveys on special-interest topics and expanded methodological instrumentation.

The 28<sup>th</sup> ANES time series study in 2008 (The American National Election Studies, 2015) has been supplemented by a coding project for open-ended responses (Krosnick et al., 2012) to various pre- and post-election questions. The topics ranged from reasons to vote for a presidential candidate, perceived reasons why a candidate won or lost the 2008 election, across the most important problems for the country and the electorate, over to (dis)likes of the competing political figures and parties among the respondents.

Like in all previous ANES studies conducted in years of presidential elections, respondents were interviewed in pre-election interviews and then re-interviewed in the two months following the election (post-election interviews), hence there was a varying number of respondents.

<sup>3</sup>Despite these benchmark collections do include data sets with text input, all inputs are provided as bag-of-words representations or similar, but **not** as full-text verbatims.

## 3 RELATED WORK

Card and Smith (2015) already investigated machine learning methods for automated coding of the ANES 2008 data. Namely, they evaluated (regularized) logistic regression models as well as recurrent neural network architectures, including long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997). As a result, they find that recurrent neural network based methods are not generally able to outperform the more "traditional" natural language processing methods, like logistic regression models combined with uni-/bigrams or additional features. An interesting conclusion they draw from their analysis is that this might be due to the limited amount of training data available for this multi-label classification task at hand. Since this is a problem statement explicitly addressed by recent transfer learning approaches, we are curious to find out whether pre-trained architectures like BERT & Co. are able to perform better on this task. Roberts et al. (2014) work on the ANES 2008 data by applying a structural topic model as a fully unsupervised approach for automated coding, which is a highly interesting strategy for previously unlabeled data sets. But since our goal is to evaluate the ability of transfer learning models (which rely on labeled data) for multi-label classification, we do not make use of this methodology.

## 4 MATERIALS AND METHODS

### 4.1 Preparation of the ANES Data

The data from the *Open Ended Coding Project*<sup>4</sup> consists of a main file in \*.xls - format which combines all verbatims<sup>5</sup> from the targeted respondents collected on the individual questions in separate spreadsheets. The codes assigned to these verbatims are stored separately in so-called *codes-files*.

Analogously to the work of Card and Smith (2015), we only use the answers to the open-ended questions *unrelated* to occupation/industry of the respondents. The topics of the questions defining the different data sets are displayed in Tab. 1. As some of the questions share the same code sets, they can be grouped into ten individual data sets comprising all of the questions on the topics mentioned in Sec. 2. With this, we follow the data preparation strategy of Card

<sup>4</sup>Publicly available under: ANES time series study and the Open Ended Coding Project

<sup>5</sup>Answers to the open-ended survey questions are referred to as *verbatims*

and Smith (2015), to keep our later results roughly comparable to their model benchmarks.

Until now, there seems to be no broadly accepted data format or structure in the social sciences regarding the storage and publication of codes assigned to individual responses to open-ended questions in surveys. Data sets seem to be structured matrix-like ad-hoc to fit an individual survey's needs.

Besides the obvious structural requirements, namely that the codes assigned to each response have to be identifiable using a particular variable (here this is provided via an "ID", alternatively designated as "caseID") and that there is a limited amount of variables which can be used for storing the code values for a single response, the internals of such data sets seem to be highly idiomatic. Another aspect which partially varies between different surveys are the codes being used for indicating that a value is missing. This in turn leads to the problem that these data sets as such are hardly usable for standard machine learning purposes without extensive preprocessing which has to reflect the individual survey's logic.

In the particular case of the ANES 2008, one has to turn to the so-called "coding report" accompanying each response-codes data set to identify the columns which contain the codes for a specific question and to understand their meaning. The pre-defined codes for each question have been manually assigned to the individual responses by professional human coders. The coding procedure has been developed after a thorough review of the ANES open-ended coding methods and a subsequent conference in December 2008<sup>6</sup> which suggested best practices.

As the sets of predefined codes belonging to individual questions cannot be used for machine learning purposes as such, we have to transform them into a useful format. In order to generate usable data sets from the files distributed by the *Open Ended Coding Project*, we exploit the notion of representing the codes, which have been assigned to each textual observation, by a binary vector.

As described previously by various authors (Tsoumakas and Katakis, 2007; Gibaja and Ventura, 2015; Herrera et al., 2016), multi-label problems can be formalized by proposing an output space  $L = L_1, L_2, \dots, L_q$  of  $q$  labels ( $q > 1$ ), which allows us to describe each observation in the data as  $(\mathbf{x}, Y)$  where  $\mathbf{x} = (x_1, \dots, x_d) \in X$  is a  $d$ -dimensional instance which has a set of labels associated  $Y \subseteq L$ . In this paper, we understand the codes assigned to each response in the data as the labels encountered in a multi-label learning problem, just as Card and Smith (2015) did pre-

viously. In order to transform the numeric codes assigned to the responses into *multi-hot encodings*, we exploit the cardinality of the code set associated with each question. This helps us to represent the labels associated to each observation by a  $q$ -dimensional binary vector  $\mathbf{y} = (y_1, \dots, y_q) = \{0, 1\}^q$  where each element is 1 if the respective label was assigned to the response and 0 otherwise.

To map the numeric codes to binary label vector elements one-to-one, we sourced the total size of each code set from the *codes*-documents enclosed with each data set. Using this information, we defined the length of the binary mapping vectors to be identical to the cardinality of the code sets. To generate multi-hot encoded label vectors for each response contained in the data sets, we designed a mapping dictionary for each code set defining which code from the current set belongs to which element in the binary vector generated for a particular response. To finally obtain the binary label vectors from the set of numeric codes associated to each observation, we transformed all data sets using a custom function which can be fed a mapping dictionary and the raw data row-by-row. The function then returns the binary label vectors of length  $q$  for each observation, where each vector element is 1 if the code mapped to this element was assigned to the response and 0 otherwise. For the latter application of machine learning methods we split the data into train and test set (90/10) using an iterative stratification method for balancing the label distributions (Sechidis et al., 2011; Szymański and Kajdanowicz, 2017a) implemented in the novel *scikit-multilearn* library for Python (Szymański and Kajdanowicz, 2017b). This represents an innovation, as such stratification has not been previously used by Card and Smith (2015). The resulting data splits are publicly available.<sup>7</sup>

## 4.2 Model Architectures

**Simple Baseline.** As a simple baseline we use a logistic regression classifier (without regularization) for *one vs. rest classification* per label and thus obtain a varying number of single models per label set. Verbatim-level averaged *fasttext*-vectors (Bojanowski et al., 2017) are used as input and one-hot vectors per label as targets. We use *nltk* (Bird et al., 2009) for a mild preprocessing of the raw verbatims, dropping punctuation, interviewer annotation and lowercasing. Then, we fit the model using the *scikit-learn* implementation (Pedregosa et al.,

<sup>6</sup>The ANES Conference on Optimal Coding of Open-Ended Survey Data took place in Dec. 2018

<sup>7</sup>Code, data sets and leaderboard available at [https://github.com/mxli417/co\\_benchmark](https://github.com/mxli417/co_benchmark).

Table 1: Overview of the prepared data sets of ANES 2008, which our analysis will be based on, and their respective topics. Additional details and descriptive statistics about the data sets can be found in Appendix 7.1.

ID	Topic	Question ID	n	#labels
1	General Election	T5, T6	238	34
2	Primary Election	T2, T3	288	29
3	Party (Dis-)Likes	C1b, C1d, C2b, C2d	4393	33
4	Person (Dis-)Likes	A8b, A8d, A9b, A9d	4672	34
5	Terrorists	S1	2100	26
6	Important Issues	Q3a1, Q3a2, Q3b1, Q3b2	8399	72
7	Office Recognition Question: Gordon Brown	J3c	2096	9
8	Office Recognition Question: Dick Cheney	J3b	2094	11
9	Office Recognition Question: Nancy Pelosi	J3a	2094	14
10	Office Recognition Question: John Roberts	J3d	2092	9

2011) in conjunction with `gensim` (Radim Rehurek, 2010) for including the `fasttext`-vectors.

**Transfer Learning Architectures.** As representatives for the class of transfer learning models we use existing `cased`<sup>8</sup> implementations of BERT-base (Devlin et al., 2018), RoBERTa-base (Liu et al., 2019) and XLNet-base (Yang et al., 2019) via `simpletransformers`, which is based on the `transformers` module (Wolf et al., 2019). The basic structure of the models is complemented by a multilabel-classification head<sup>9</sup>. The used loss function is `BCEWithLogitsLoss` from `pytorch` *per node* in order to account for the multi-label structure of the targets. We do not intend to perform excessive tuning of hyperparameters, but rather want to evaluate the performance of these models when used "out-of-the-box" for a much more difficult task than the common ones. This approach is also largely in line with recent works extending BERT to multi-label problems (Lee and Hsiang, 2019; Chang et al., 2019). All models were fine-tuned on the data sets for three epochs with a maximum sequence length of 128 tokens and a batch size of eight sequences. (Peak) learning rate for fine-tuning was set to  $2e-05$  for every model.

### 4.3 Evaluation Metrics

Generally, metrics commonly used for the evaluation of machine learning methods in binary or multi-class classification tasks cannot be used for multi-label learning without some further considerations (Tsoumakas and Katakis, 2007). This is mainly due to the fact that the performance of a given classifier should be evaluated over all labels and the partial correctness of a prediction must be taken into account.

<sup>8</sup>Since RoBERTa only exists in a `cased` version, we had to choose the other models analogously.

<sup>9</sup>For implementation details of this head see <https://github.com/ThilinaRajapakse/simpletransformers>

Thus, we here utilize a set of multi-label evaluation metrics reported in overview articles by different authors (Tsoumakas and Katakis, 2007; Sorower, 2010; Gibaja and Ventura, 2014, 2015; Herrera et al., 2016) to assess various aspects of the performance of the classifiers we investigate.

For the following, we resume the previous notation. Let us assume that we have a multi-label test set  $T = (\mathbf{x}_i, Y_i) | 1 \leq i \leq n$  with  $n$  instances and different label sets  $Y_i$ , representing the ground truth, at our disposal. Further, let  $P_i$  be the set of predicted labels for a given observation.

First, we will report the widely known  $F_1$  score, which is the harmonic mean of *Precision* and *Recall*

$$F_1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \quad (1)$$

We report the micro- and macro-averaged versions of this score, as the  $F_1$  score is a binary evaluation measure and one needs to choose an averaging approach in the multi-label case. By doing so, different performance aspects can be investigated (Gibaja and Ventura, 2015). Micro-averaging mainly tends to summarize the classifier performance on the most common categories, whereas macro-averaging tends to report performance on the rare categories of the test set. Values towards 1 are better, the minimum value is 0.

Additionally, we also report the sample-based  $F_1$  score as this is also the central metric Card and Smith (2015) use and report in their paper<sup>10</sup>. This version of the  $F_1$  score can be formally described as:

$$F_1^{\textit{sample}} = \frac{1}{N} \sum_{i=1}^N \frac{2|Y_i \cap P_i|}{|Y_i| + |P_i|} \quad (2)$$

(cf. Gibaja and Ventura (2014)) where  $N$  is the total number of samples in the test set.

Second, we report the *subset accuracy*, often also referred to as *exact match ratio*. It computes the fraction of instances in the data for which the predicted

<sup>10</sup>Note that they did not use the same notation, but essentially used the same metric described in a vectorized form.

Table 2: Model performances (measured as micro- and macro-averaged  $F_1$ -scores) for all considered architectures. Results are displayed separately for each data set with the best performance per data set in bold. We report  $F_1^{sample}$  to ensure comparability to the results reported by Card and Smith (2015).

Dataset-ID		1	2	3	4	5	6	7	8	9	10
$n$		238	288	4393	4672	2100	8399	2096	2094	2094	2092
#labels		34	29	33	34	26	72	9	11	14	9
$F_1^{sample}$	Baseline	0.44	0.51	0.57	0.54	0.68	<b>0.88</b>	0.92	0.95	0.90	0.91
	BERT	0.00	0.02	0.44	0.35	0.41	0.79	0.94	0.95	0.91	0.93
	RoBERTa	0.00	0.00	0.56	0.55	0.57	0.85	0.95	0.97	<b>0.93</b>	0.94
	XLNet	0.00	0.00	0.54	0.58	0.55	0.86	<b>0.96</b>	<b>0.98</b>	0.91	0.92
	Card and Smith (2015)	<b>0.55</b>	<b>0.67</b>	<b>0.71</b>	<b>0.71</b>	<b>0.81</b>	0.86	0.94	0.96	<b>0.93</b>	<b>0.96</b>
$F_1^{micro}$	Baseline	<b>0.40</b>	<b>0.48</b>	0.53	0.51	0.61	0.84	0.89	0.93	0.85	0.90
	BERT	0.00	0.03	0.51	0.44	0.46	0.79	0.94	0.95	0.91	0.93
	RoBERTa	0.00	0.00	<b>0.60</b>	0.60	<b>0.62</b>	<b>0.85</b>	<b>0.96</b>	<b>0.97</b>	<b>0.94</b>	<b>0.95</b>
	XLNet	0.00	0.00	0.59	<b>0.61</b>	0.61	<b>0.85</b>	<b>0.96</b>	<b>0.97</b>	0.90	0.93
$F_1^{macro}$	Baseline	<b>0.23</b>	<b>0.29</b>	<b>0.33</b>	<b>0.34</b>	<b>0.47</b>	<b>0.46</b>	<b>0.62</b>	0.51	<b>0.56</b>	<b>0.71</b>
	BERT	0.00	0.01	0.11	0.16	0.12	0.09	0.47	0.40	0.39	0.58
	RoBERTa	0.00	0.00	0.18	0.26	0.21	0.14	0.51	0.51	0.44	0.58
	XLNet	0.00	0.00	0.20	0.27	0.21	0.16	0.58	<b>0.53</b>	0.43	0.66

labels *exactly* match their corresponding true labels. This is a very harsh metric, as it does not distinguish between partially and completely incorrect predictions. It is defined as:

$$subset\ accuracy = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(P_i = Y_i) \quad (3)$$

Next, we report the *Label Ranking Average Precision* (LRAP). This metric computes the fraction of labels ranked above a certain label  $\lambda \in Y_i$  which belong to  $Y_i$ , averaged across all observations (Gibaja and Ventura, 2015). For this, a function  $f: X \times Y \rightarrow \mathbb{R}$  is generated by a label-ranking algorithm which orders all possible labels for a given instance  $\mathbf{x}_i$  by their relevance (Gibaja and Ventura, 2014). If a given label  $\lambda' \in Y_i$  is ranked higher than a another label  $\lambda \in Y_i$ , then  $f(\mathbf{x}_i, \lambda') > f(\mathbf{x}_i, \lambda)$  must hold. In the following we consider  $\hat{f}_\lambda$  to be a function which returns the ranking for a given label  $\lambda$ , generated by the used ranking algorithm. Here, the higher the obtained metric results are, the better. The best achievable value is 1. LRAP is defined as (Gibaja and Ventura, 2014):

$$LRAP = \frac{1}{N} \sum_{i=1}^N \frac{1}{|Y_i|} \sum_{\lambda \in Y_i} \frac{|\{\lambda' \in Y_i | \hat{f}_{\lambda'} \leq \hat{f}_\lambda\}|}{\hat{f}_\lambda} \quad (4)$$

The LRAP favors classifiers which can rank the relevant labels associated with each observation higher than the irrelevant ones.

## 5 RESULTS

We report all of the above mentioned metrics for the baseline model as well as for the three mentioned pre-

trained architectures on the test set. The results will be structured as follows: In Tab. 2 we report macro- and micro-averaged  $F_1$  scores, additionally the sample-based  $F_1$  scores (cf. Card and Smith 2015) will be reported as well. Tab. 3 shows the label ranking average precision LRAP and the *subset accuracy*.

Considering the  $F_1^{sample}$  scores from Tab. 2, it becomes clear that all used models can hardly outperform the previous best results. Note that the best model from Card and Smith (2015) on almost all data sets has been a thoroughly tuned logistic regression model with a battery of different features. Overall, the best logistic regression model has outperformed even much more advanced architectures in 7 out of 10 cases, establishing that this kind of model can handle multi-label text classification problems surprisingly well. In line with this, we observe that our baseline can beat the transfer learning architectures on 5 out of 10 data sets. Only RoBERTa and XLNet can beat the previous best results on two data sets by a small margin. On all other data sets the previously set benchmark results remain largely unchallenged.

When focussing on the  $F_1^{micro}$  measure, we can see that the more advanced models, especially RoBERTa and XLNet, outperform the baseline as soon as the data set size gets bigger, even if they sometimes demonstrate only a slightly better performance. BERT still performs relatively poorly, and even gets beaten by the baseline on five out of ten data sets. RoBERTa also shows only slightly better performance than the baseline on the data set 5 containing the question on terrorism and the data set 6 on Important Issues. On the remaining data sets, how-

Table 3: Model performances (measured as *LRAP* and *subset accuracy*) for all considered architectures. Results are displayed separately for each data set with the best performance per data set in bold.

Dataset-ID		1	2	3	4	5	6	7	8	9	10
<i>LRAP</i>	Baseline	<b>0.59</b>	<b>0.65</b>	<b>0.70</b>	<b>0.70</b>	<b>0.75</b>	<b>0.93</b>	<b>0.95</b>	<b>0.98</b>	0.92	<b>0.95</b>
	BERT	0.09	0.10	0.41	0.32	0.40	0.71	0.94	0.95	0.90	0.93
	RoBERTa	0.09	0.09	0.51	0.49	0.55	0.79	<b>0.95</b>	0.97	<b>0.93</b>	0.94
	XLNet	0.09	0.09	0.49	0.52	0.53	0.80	<b>0.95</b>	0.97	0.90	0.92
<i>subset acc.</i>	Baseline	0.00	<b>0.10</b>	0.20	0.17	<b>0.35</b>	<b>0.70</b>	0.80	0.89	0.76	0.80
	BERT	0.00	0.00	0.16	0.08	0.20	0.41	0.89	0.90	0.81	0.87
	RoBERTa	0.00	0.00	<b>0.22</b>	0.20	0.32	0.54	0.91	<b>0.94</b>	<b>0.87</b>	<b>0.89</b>
	XLNet	0.00	0.00	<b>0.22</b>	<b>0.22</b>	0.31	0.58	<b>0.92</b>	<b>0.94</b>	0.80	0.87

ever, it can clearly outperform the baseline. XLNet also mostly outperforms the baseline, with the exception of the data set concerning terrorism. On the very small and thus very challenging data sets 1 and 2 which contain questions on the General and Primary Election outcomes, the baseline model still is the best.

Finally, when considering the  $F_1^{macro}$  score, we observe that the baseline model is the single best model across almost all data sets. Only for data set 8, the larger RoBERTa and XLNet can match or outperform it. While this might be quite surprising, it proves again that a binary relevance approach with a logistic regression as a base learner can be a quite competitive model – which is exactly the same finding Card and Smith (2015) have reported.

Regarding *LRAP* (cf. Tab. 3), RoBERTa and XLNet can partially match the baseline model especially on the last four data sets, which have a small label set and are reasonably large. But XLNet and RoBERTa also hardly outperform the baseline on all remaining data sets, which makes the baseline model a powerful ranking algorithm. BERT, however, cannot beat the baseline at any of the data sets. For the strict measure *subset accuracy* the baseline is not a strictly superior competitor, as it outperforms the more advanced models only on 3 out of 10 data sets. This is also why it is important to compare several evaluation metrics in multi-label classification, as each metric focuses different performance characteristics (Nam, 2019). Unfortunately, Card and Smith (2015) have not provided any results beyond the  $F_1^{sample}$  metric.

After these comparisons we conclude that concerning data sets 1 and 2, which contain 238 and 288 observations respectively, BERT, RoBERTa and XLNet cannot obtain any results above zero. Additionally, these models outperform the baseline only marginally on the data sets regarding the Party (Dis)Likes, Person (Dis)Likes and the Office-Recognition-Question for Dick Cheney. Nonetheless, they can outperform the baseline as soon as the data

sets get larger and the label sets remain relatively small.

## 6 DISCUSSION

Transfer learning has, in this specific use case, not turned out to be a strong alternative compared to previous research. BERT, RoBERTa and XLNet can not generally outperform previous best results obtained on the same data. Additionally, we observed just like the previous authors that a binary relevance approach with logistic regression can be a strong competitor, sometimes even outperforming advanced models. On small data sets, however, no model achieved good results with respect to the subset accuracy, our hardest metric. This is most certainly due to the size, as the data does not contain much information for automated classifiers to learn from. In this case, relying on hand-coding by humans might still be a good option.

Our findings are somewhat contrary to previously reported results, where BERT was used quite successfully in multi-label classification (Adhikari et al., 2019; Chang et al., 2019; Lee and Hsiang, 2019), even yielding new SOTA results. The data sets these authors have used to train their models, however, were much larger than the ones we can utilize here. As noted previously, we try to generate a benchmark regarding the usability of these models in the context of scarce data, which is common in the social sciences. In the light of the good performance of the baseline model, the bigger models also might not be the best choice if computational efficiency is the goal. As social scientists typically do not have unlimited computing power at their disposal, a model which can be trained to obtain reasonable levels of, for example, subset accuracy, in a short amount of time might be especially interesting for future research. Additionally, this model also can handle smaller data sets significantly better and does not break down on bigger

ones. This might be an indicator to look at smaller, more problem-specific algorithms like feature-based transfer learning to advance the research on automatic coding in the future.

## 7 CONCLUSION

In this work, we provided an extension to the collection of commonly used benchmark data sets used for evaluation transfer learning models for NLP. The full-text data set encompasses a different task than most of the others and thus widens the opportunities for carefully evaluating pre-trained models on a different kind of challenge. Furthermore we propose a unified pre-processing of the data set going along with a fixed train-test split enabling a valid comparison against our baselines. We evaluated the performance of state-of-the-art transfer learning models on the ANES 2008 data set and compared them to a simple baseline model. Our comparison illustrates that, despite the extremely good performances of those models on binary, multi-class and previous multi-label classification tasks, there is still a lot of room for improvement concerning the performance on challenging multi-label classification tasks on small to mid-sized data sets.

## ACKNOWLEDGEMENTS

We want to express our sincere gratitude to Christian Heumann for his guidance and support during the process of this research project. We would like to thank Jon Krosnick and Matt Berent for their insightful explanations via e-mail regarding the *Open Ended Coding Project*. This helped us to develop a better understanding for the initial data format. A special thanks also goes to Dallas Card for his explanations regarding the data splits from Card and Smith (2015).

## REFERENCES

- Adhikari, A., Ram, A., Tang, R., and Lin, J. (2019). Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.
- Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. Safari Books Online. O'Reilly Media Inc, Sebastopol.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Card, D. and Smith, N. A. (2015). Automated coding of open-ended survey responses.
- CESSDA Training Team (2020). Cessda data management expert guide.
- Chang, W.-C., Yu, H.-F., Zhong, K., Yang, Y., and Dhillon, I. (2019). X-bert: extreme multi-label text classification using bidirectional encoder representations from transformers. *arXiv preprint arXiv:1905.02331*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Gibaja, E. and Ventura, S. (2014). Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(6):411–444.
- Gibaja, E. and Ventura, S. (2015). A tutorial on multilabel learning. *ACM Computing Surveys (CSUR)*, 47(3):1–38.
- Herrera, F., Charte, F., Rivera, A. J., and Del Jesus, M. J. (2016). Multilabel classification. In *Multilabel Classification*, pages 17–31. Springer.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hoyle, L., Vardigan, M., Greenfield, J., Hume, S., Ionescu, S., Iverson, J., Kunze, J., Radler, B., Thomas, W., Weibel, S., and Witt, M. (2016). Ddi and enhanced data citation. *IASSIST Quarterly*, 39(3):30.
- Inter-University Consortium For Political And Social Research (ICSPR) (2012). Guide to social science data preparation and archiving: Best practice throughout the data life cycle.
- Krosnick, J. A., Lupia, A., and Berent, M. K. (2012). 2008 open ended coding project.
- Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. (2017). Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.
- Lee, J.-S. and Hsiang, J. (2019). Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv:1906.02124*.
- Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pre-training approach. *arXiv preprint arXiv:1907.11692*.
- Lupia, A. (2018a). Coding open responses. In Vannette, D. L. and Krosnick, J. A., editors, *The Palgrave Handbook of Survey Research*, pages 473–487. Springer International Publishing, Cham.
- Lupia, A. (2018b). How to improve coding for open-ended survey data: Lessons from the anes. In *The Palgrave Handbook of Survey Research*, pages 121–127. Springer.
- Mencia, E. L. and Fürnkranz, J. (2008). Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Joint European Conference*

on *Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer.

Nam, J. (2019). *Learning Label Structures with Neural Networks for Multi-label Classification*. PhD thesis, Technische Universität.

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

Radim Rehurek, P. S. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., Albertson, B., and Rand, D. G. (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082.

Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of multi-label data. In Gunopulos, D., Hofmann, T., Malerba, D., and Vazirgiannis, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 145–158, Heidelberg. Springer.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 18:1–25.

Szymański, P. and Kajdanowicz, T. (2017a). A network perspective on stratification of multi-label data. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 22–35.

Szymański, P. and Kajdanowicz, T. (2017b). A scikit-based python environment for performing multi-label classification. *arXiv preprint arXiv:1702.01460*.

The American National Election Studies (2015). *ANES 2008 Time Series Study*. Inter-university Consortium for Political and Social Research [distributor].

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Vardigan, M., Heus, P., and Thomas, W. (2008). Data documentation initiative: Toward a standard for the social sciences. *International Journal of Digital Curation*, 3(1):107–113.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized au-

to-regressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Züll, C. (2016). Open-ended questions. *GESIS Survey Guidelines*, 3.

## APPENDIX

### 7.1 Pre-processed Data Set

Table 4: Multi-label descriptive statistics for our data preparation approach.

ID	Word count		Avg. Words/ Verbatim	-Cardinality	Label -Density	-PowerSet Size	PowerSet/ Examples		Observations per Label	
	Total	Unique					Min.	Avg.	Max.	
1	3775	872	15.86	2.895	0.085	238	1.00	1	20.26	73
2	3176	752	11.03	2.205	0.076	288	1.00	1	21.90	114
3	60405	5046	13.75	2.291	0.069	4393	1.00	2	305.00	1507
4	67659	5136	14.48	2.397	0.070	4672	1.00	1	329.32	1549
5	26502	2442	12.62	1.947	0.075	2100	1.00	8	157.27	618
6	37652	3548	4.48	2.329	0.032	8399	1.00	1	271.64	8398
7	9512	742	4.54	1.374	0.153	512	0.24	5	319.89	1363
8	6711	576	3.20	1.204	0.109	2048	0.98	1	229.18	1384
9	10378	720	4.96	1.369	0.098	2094	1.00	3	204.71	899
10	9157	762	4.38	1.337	0.149	512	0.24	10	310.78	1232