Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

GermEval 2019 Task 1: Hierarchical Classification of Blurbs

Steffen Remus, Rami Aly, Chris Biemann

Language Technology Group
Department of Informatics
Universität Hamburg, Germany
{remus,5aly,biemann}@informatik.uni-hamburg.de

Abstract

This paper presents the setup and outcome of the GermEval-2019 Task 1: Hierarchical Classification of Blurbs. A blurb is a short, occasionally advertorial, description of a book. The shared task consists of two subtasks: Task **A**) classification of blurbs exclusively into the most general categories, which can be considered to be a multi-label classification task, and Task **B**) hierarchical classification of blurbs into the entire hierarchy of categories, spanning a total of 343 different categories and sub-categories. During the test period, ten teams submitted 17 valid system solutions for Task A, and eight teams submitted 16 system solutions for Task B. For Task A, the best submission achieved a micro-F_1 score of 0.867, and for Task B the best submission achieved a micro-F_1 score of 0.677.

1 Introduction

Text classification (TC), as a sub-discipline in natural language processing (NLP), is an established task where many datasets for many target domains and challenges exist. Spam classification is probably the most well-known application of text classification algorithms. Here, the task is to classify messages (emails or short text messages) into two classes: spam (advertisements or any kind of harassment messages), or ham (relevant messages; Gómez Hidalgo et al., 2006[1]). Due to the nature of this task and the fact that this resolves to binary text classification, it can be considered being solved with accuracy scores reaching 98+%, see e.g. (Taheri and Javidan, 2017). However, as more and more data become digitally available and people's time and convenience are growing in priority, the demand for more, and finer-grained categories increases. Multi-class text classification gathered attention in this space (e.g. with the 20 Newsgroups dataset[2]), here the task is to classify an email (text and metadata) into one of 20 possible categories. As a next step, the multi-class text classification problem has been developed into a multi-label text classification problem, where a single sample can have one or multiple class labels. One of the popular datasets in this domain is the Reuters-21578 dataset[3] (Lewis, 1992) which was superseded by the RCV1 dataset[4] (Reuters Corpus Volume 1; Lewis et al., 2004), implementing a hierarchical structure on the classes. In hierarchical multi-label classification (HMC), labels are organized in a structured hierarchy, i.e. certain label combinations are irrelevant and should never be classified in conjunction (Silla and Freitas, 2011).

Hierarchical multi-label classification is not an entirely new challenge in the area of natural language processing (Sun and Lim, 2001; Silla and Freitas, 2011), but with the increase of available data, especially on the web, the desire for more specific and specialized hierarchies increases. To cover this desire, and to foster research for algorithms dealing with hierarchically organized classes for the German Language in a real-world scenario, we present the *GermEval-2019 Task 1: Hierarchical Classification of Blurbs*, which includes two subtasks, where automatic systems have to infer: **A**) the **most general categories** of a book described by a blurb, and **B**) the **entire**

[1] http://dcomp.sor.ufscar.br/talmeida/smsspamcollection/

[2] http://qwone.com/~jason/20Newsgroups/

[3] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[4] http://www.daviddlewis.com/resources/testcollections/rcv1/

	Task A	Task B
#Teams:	10	8
#Submissions:	17	16
Best Team:	EricssonResearch	TwistBytes
Best Micro-F_1:	0.867	0.6767
Impr. over Baseline:	0.067	0.1428

Table 1: Quantitative details of submissions.

set of categories in the class hierarchy.[5,6] Since a sample can belong to multiple classes on the same level, Task A can be considered as a standard multi-label classification task and a sub-problem of Task B, which is a hierarchical multi-label classification task. We compiled a hierarchical dataset of German blurbs by crawling the web pages of a major publisher and taking care of proper data cleaning and preparation.[7] The details of the entire process, as well as various statistics, can be found in Section 3. For the shared task, we allowed three system submissions per team where eventually ten teams submitted 17 valid system solutions for Task A, and 16 valid system solutions were submitted by eight teams for Task B. Quantitative details of the test-phase submissions can be found in Table 1.

2 Prior Work

Text Classification Datasets:

The probably most well-known dataset with a hierarchical class label structure is the RCV1 (Reuters Corpus Volume 1; Lewis et al., 2004) dataset. It consists of roughly 800K documents categorized into several hierarchically structured category sets. However, the access to the dataset is limited and not freely usable by e.g. companies due to licensing. Lewis et al. (2004) distribute a term-document matrix where it has been ensured that the original data cannot be reconstructed. Therefore, many different variations of the original dataset have been created and used, and despite the wide acceptance of the dataset and extensive usage, it is difficult to directly compare results presented in scientific work due to the lack of availability of the standardized version.

Kowsari et al. (2017) introduced a hierarchically structured dataset for English, with a maximum depth of two, called the Web of Science Dataset: WOS-11967, WOS-46985 and WOS-5736 with 35, 134 and 11 categories and 7, 7 and 3 top-level categories respectively. However, in this dataset, every sample consists of exactly one parent-child label, which ultimately results in a single-label multi-class problem on the more specific category. This highly limits the diversity and complexity of the dataset and the underlying hierarchy. Several other large-scale datasets have been presented, e.g. (Kim et al., 2019; Mencía and Fürnkranz, 2010; Partalas et al., 2015). Some of these datasets consist of an extensive number of classes, up to several thousand. The classification of these datasets carry their very own challenges and are thus not further discussed here. In special application domains, such as the biomedical domain, more and more works include hierarchical structures in their data: e.g. Baker et al. (2015) introduced an annotated dataset based on the hallmarks of cancer (Baker et al., 2017) with a total of 37 classes and a hierarchy depth of 3 levels; Larsson et al. (2017) compiled a dataset for chemical risk assessment with a 32 classes and 5 levels.

Many freely accessible hierarchical datasets for the German language exist, however, no benchmark dataset has been established. For example, the OAI Protocol for Metadata Harvesting is a protocol designed to share metadata of catalogs and publications. However, the minimal requirements for expressing valid records are fairly loose and the practices of metadata management wildly differ across repositories. Attempts have been made to normalize OAI metadata records according to the hierarchical library taxonomy (Waltinger et al., 2009), called the Dewey Decimal Classification system. Multiple datasets of German patent collections have been created to classify these documents into the IPC taxonomy (Fall et al., 2004; Tikk et al., 2005).

HMC Approaches:

In text classification without hierarchical structures, neural architectures, especially *Convolutional Neural Networks* (CNNs) and different types of *Recurrent Neural Networks* (RNNs) (Goodfellow et al., 2016; Kim, 2014), most notably long short-term memory units (LSTMs,

[5] GermEval is a series of shared task evaluation campaigns that focus on Natural Language Processing for the German language. The workshop is held in conjunction with the Conference on Natural Language Processing KONVENS 2019 in Erlangen/Nürnberg.

[6] https://competitions.codalab.org/competitions/20139

[7] We crawled the websites with the consent of the Random House publisher group.

Hochreiter and Schmidhuber, 1997) have shown to be highly effective. Cerri et al. (2014) use concatenated multi-layer perceptrons (MLP), where each MLP is associated with one level of the class hierarchy. In contrast, classifier chains (Read et al., 2011) employ binary classifiers for each category and propagate their predictions as a feature to the classifier for the child categories. However, this method is computationally expensive. Kowsari et al. (2017) use multiple concatenated deep learning architectures (CNN, LSTM, and MLP) for the WOS dataset – with a very shallow hierarchy and a fixed number of classes per example (one class label for each of the two hierarchy levels). Traditional classification approaches, such as e.g. KNN, Naïve Bayes or SVM, appear to fail to generalize adequately for large hierarchies (Kowsari et al., 2017). Summarizing, hierarchical multi-label classification brings research-worthy challenges, which motivated the conduction of this shared task.[8]

3 Dataset

In the following, we describe the preparation steps of the dataset, which are strongly in line with Aly et al. (2019).

3.1 Compiling the Dataset

The dataset is compiled using the openly available data of the (Bertelsmann) Random House (RH) webpage[9]. Random House is worldwide the largest publisher group and thus hosts an enormous body of books.

The German webpages of RH provide various meta information of books, such as a short description (the blurb), authorship information, title of the book, etc. (c.f. Figure 1). With the permission of the German RH division, we crawled[10] the book listings, parsed the HTML code[11] and collected the following information that we considered to be relevant:

- title
- author(s)
- URL
- ISBN
- date of publication
- genres, i.e. categories
- info text, i.e. the blurb content

Other information such as *about the author*, or *reader's ratings* were ignored. The blurb of a book can be considered to be a short incentive description, which is occasionally advertorial (i.e. advertising and editorial) and thus clearly distinctive to a summary. Blurbs aim to bestir a potential reader to buy and read the book, they are thus designed to occasionally contain advertorial content. Each collected blurb can be considered unique, however, they might appear in similar forms, e.g. for books that are part of a series or are being republished as a new edition due to their success. Due to the extraction process of the sometimes noisy web data, anomalies such as missing author, missing blurb or incorrect publication date occurred infrequently for about 1% of the collected data and were thus accepted and kept in the dataset.

3.2 Category Refinement

The per-book extracted categories are lists of genres connected with their ancestor genres. Each book is thus categorized into a hierarchy. Still, this hierarchy contains ambiguities caused by the assignment of identical names to different categories allowing the formation of cycles as well as children to have multiple parents, e.g. *Science Fantasy* occurs as a subcategory of *Science Fiction* and *Fantasy*. Thus, we automatically renamed ambiguous categories by concatenating the category name to its parent's category name, and manually refined the extracted hierarchy further, which results in a tree-like categorical structure. Further, we manually checked all relations and merged or removed similar labels and removed categories that capture properties that do not rely on content but the shape or form of a book, e.g. categories such as *audiobook*, *ebook*, *hardcover*, *softcover*, etc. were removed. Finally, samples that have assigned category combinations that appear less than five times were also removed from the dataset.

3.3 Dataset Properties

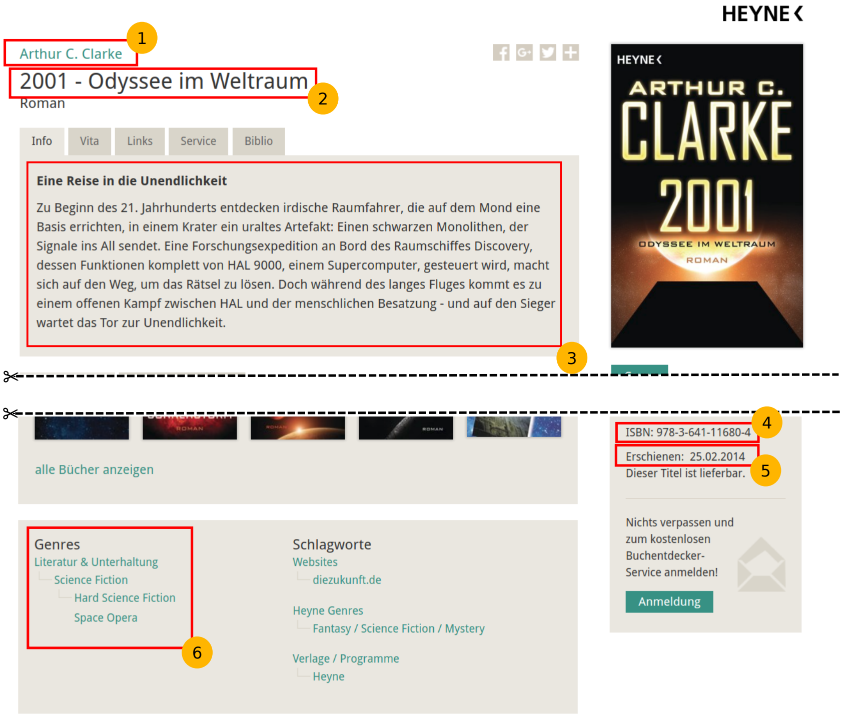
The dataset follows the requirements as described in (Lewis et al., 2004): First, every book is as-

[8] The official webpage of the shared task and respective data can be found at https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/germeval-2019-hmc.html.

[9] https://www.randomhouse.de/

[10] We crawled the webpages with Scrapy (https://scrapy.org/).

[11] XPath and CSS where used to find and extract the necessary information.



HEYNE❮

Arthur C. Clarke

2001 - Odyssee im Weltraum

Roman

Info Vita Links Service Biblio

Eine Reise in die Unendlichkeit

Zu Beginn des 21. Jahrhunderts entdecken irdische Raumfahrer, die auf dem Mond eine Basis errichten, in einem Krater ein uraltes Artefakt: Einen schwarzen Monolithen, der Signale ins All sendet. Eine Forschungsexpedition an Bord des Raumschiffes Discovery, dessen Funktionen komplett von HAL 9000, einem Supercomputer, gesteuert wird, macht sich auf den Weg, um das Rätsel zu lösen. Doch während des langes Fluges kommt es zu einem offenen Kampf zwischen HAL und der menschlichen Besatzung - und auf den Sieger wartet das Tor zur Unendlichkeit.

alle Bücher anzeigen

ISBN: 978-3-641-11680-4

Erschienen: 25.02.2014

Dieser Titel ist lieferbar.

Nichts verpassen und zum kostenlosen Buchentdecker-Service anmelden!

Anmeldung

Genres
- Literatur & Unterhaltung
  - Science Fiction
    - Hard Science Fiction
    - Space Opera

Schlagworte
- Websites
  - diezukunft.de
- Heyne Genres
  - Fantasy / Science Fiction / Mystery
- Verlage / Programme
  - Heyne

Figure 1: Snippet of website the data was collected from. The specific parts are highlighted in red boxes. Numbers indicate specific parts: (1) author name(s), (2) title, (3) blurb, (4) ISBN, (5) release date, (6) book's categories, displayed in a tree structure according to the underlying hierarchy. [The screenshot was taken in October 2018.]

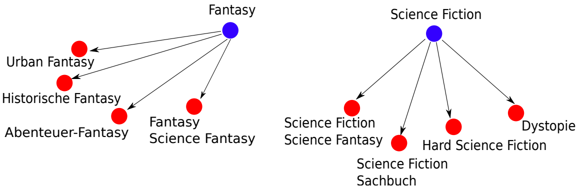


Figure 2: Excerpt of the hierarchy of categories. Colors indicate different levels in the hierarchy. The full hierarchy can be found in (Aly, 2018, p. 58).

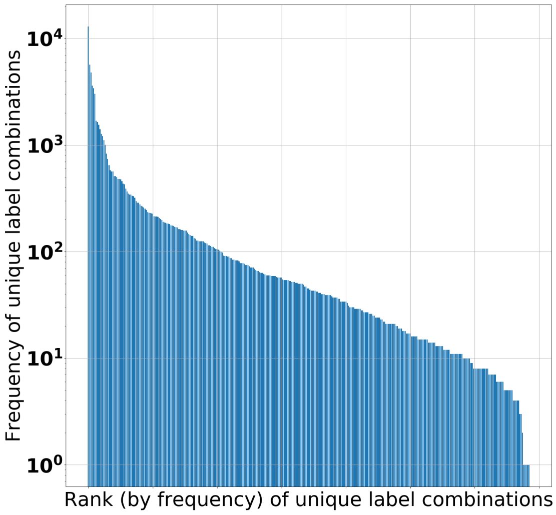


Figure 3: Frequency of category combinations (y-axis) in the entire dataset sorted by frequency rank (x-axis).

signed at least one category, and second, every parent category in the path to the most general category of a book's most specific category is transitively assigned to it as well. In the dataset, the specified labels and the transitively assigned labels are distinguishable with the XML property `label` (value = `true` for most the specific label). Note that the most specific category of a book is not necessarily a leaf category in the hierarchy. For instance, the most specific category of a book could be *Children's Books*, although further child categories, such as *Middle-Grade books*, exist.

Figure 4 shows the frequency distribution of unique category combinations sorted by frequency rank. As expected, few label combinations appear often and many label combinations appear rarely. The distribution of labels remains highly diverse with a total of 484 unique category combinations. Table 2 lists further important quantitative characteristics of the collected data such as the number of categories on each level of the hierarchy, etc.

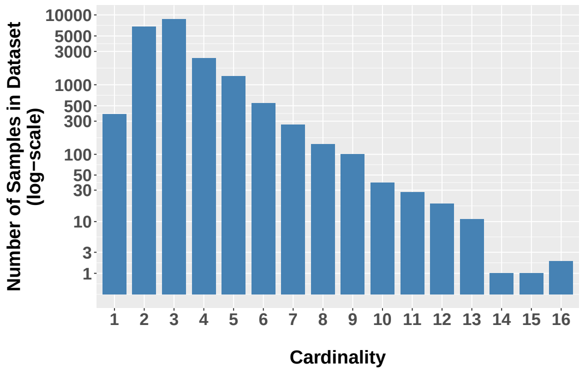


Figure 4: Distribution of the category cardinality per sample in the entire dataset.

For the task, we divided the dataset into three subsets: 70% training, 10% development and 20% test set (±0.2% respectively). The dataset was split randomly with the constraint that every category in the development and test set occurs at least once in the training set. Additionally, maximally 2% of categories in the development and test set occur less than three times in the training set. While the test set is only used for the final evaluation of each system, the development set was used for benchmarking during the first evaluation phase. During the entire runtime of the task, participants were able to compare the performances of their systems via the CodaLab leaderboard for the development set. For the final evaluation phase, the development set labels have been supplied to the participants to allow a larger training set, and the CodaLab leaderboard was disabled for test set prediction submissions to avoid optimization on the test set.

4 Task Definition

The shared task contains two subtasks:

Task A: The task is to classify German books into one or multiple top-level categories. It can thus be considered a standard multi-label classification task. In total, there are eight top-level classes that can be assigned to a book: *Literatur & Unterhaltung (Literature & Entertainment)*, *Ratgeber (Counsel)*, *Kinderbuch & Jugendbuch (Books for Children and Young Adult Readers)*, *Sachbuch (Nonfiction)*, *Ganzheitliches Bewusstsein (Holistic Awareness)*, *Glaube & Ethik (Belief & Ethics)*, *Künste (Arts)*, *Architektur & Garten (Architecture & Gardening)*. The label distribution of these eight classes is highly imbalanced (cf. Figure 5).

# Samples	$20,784$
Average blurb length in tokens	94.67
Total number of categories	343
# Categories on level:	
1	8
2	93
3	242
# leaf nodes on level:	
1	0
2	51
3	242
Average branching factor	6.7 ± 4.97
Average branching factor on level:	
1	11.63 ± 6.39
2	5.76 ± 4.12
# Samples with labels of category on level:	
1	$20,784$
2	$20,406$
3	$11,117$
# Samples w/ cardinality (tlc*):	
1	$19,422$
2	$1,260$
3	97
4 (maximum cardinality)	5
# Samples w/ cardinality:	
see Figure 4 (maximum $= 16$*)*	
Average cardinality (tlc*)	1.07 ± 0.28
Average cardinality	3.11 ± 1.37
# Distinct label combinations	484

Table 2: Quantitative characteristics of the dataset (*tlc: top-level-categories).

Task B: The second task is a hierarchical multi-label classification task where all categories of the hierarchy have to be assigned to a book. In total, 343 different classes are hierarchically structured, hence, not all combinations of categories are valid as defined by the hierarchy.

Submission Setup: The entire submission process was organized within the framework of a CodaLab competition[12]. We limited the number of system submissions to three per team. The data release cycle went in three phases: In the first phase only a limited number of samples was released to familiarize with the structure of the dataset; in the second phase the training set with labels and the development set without labels were released and participants were able to submit their solutions for the development set to the CodaLab website; the

[12] https://competitions.codalab.org/

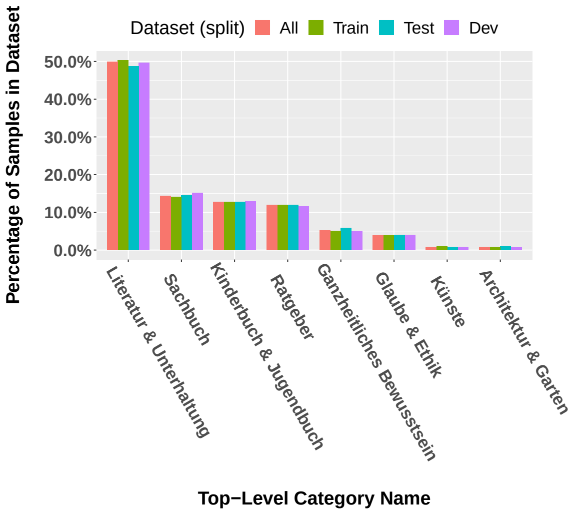


Figure 5: Top-level sample distribution.

third phase is the final test phase where the test set samples without labels and the labels for the development set samples were provided.

5 Systems

5.1 Organizer Systems

Baseline: SVM As a baseline method, we implement a traditional, non-hierarchical classifier using the *local approach* as described by Silla and Freitas (2011). We chose to use a linear SVM (Cortes and Vapnik, 1995) since it yielded good results in preliminary experiments. We exclusively use the blurb of a book to create features for the SVM and decided on minimal preprocessing, i.e. tokenization is performed using spaCy[13] and stop words – as defined by spaCy – have been filtered. We then created a bag-of-word representation of unigrams and bigrams. Since the SVM is a binary classifier, we opted for a one-vs-all multi-label classification scenario, which was implemented using the scikit-learn library[14]. We use the standard value for the hyperparameter $C = 1$ and did not fine-tune it. Because predictions by independent classifiers do not necessarily lead to valid combinations as defined by the underlying hierarchy, we apply a post-processing step where we add missing parents of each predicted child label – recap that every child has an unambiguous parent. This process provides hierarchy-consistent label combinations but might lead to incomplete combinations because we do not add child labels for inner category nodes.

#Primary capsules	100
Convolution window size	50
Dimension of primary capsules	8
Dimension of class. capsules	8
Optimizer	Adam (Kingma and Ba, 2014)
Learning rate	0.002
#Epochs	10

Table 3: Hyper-parameter settings of the capsule network as found by non-exhaustive search.

Contender: Capsule Networks Capsule networks have recently been shown to have advantages over traditional neural networks when confronted with structurally diverse categories and complex label co-occurrences (Aly et al., 2019; Zhao et al., 2018). For this reason, and the fact that the dataset is inherently unbalanced (as illustrated in Figure 3), we decided to employ a capsule network architecture from our previous work as a *contender system* for comparative reasons and out-of-competition. For the input, we tokenize the fields containing texts (title, author, and blurb) with spaCy and concatenate them. Tokens that appear only once in the dataset are replaced with a special unknown-token word. The sequence length of has been limited to 100 tokens. We initialize an embedding layer with pre-trained fastText embeddings[15] provided by Bojanowski et al. (2017) and adjust them during training. The structure of the capsule network follows tightly the implementation by Aly et al. (2019): Similar to CapsNet1 in (Xiao et al., 2018), our proposed system consists of four layers and every category in the hierarchy is associated with one class capsule in the network. As a post-processing step, we apply the same correction procedure as described above. Further hyper-parameter settings can be found in Table 3.

5.2 Submitted Systems

This section aims to give a quick overview of the different approaches used by the various teams for Task A and B, a short overview can be found in Table 4. We observe that the applied approaches can be grouped into two major groups, i.e. one focusing on the *local approach* where each node of the hierarchy is classified independently, here, mainly traditional classifiers are used, and one using the *global approach* where nodes are classified jointly

[13] https://spacy.io/

[14] https://scikit-learn.org

[15] https://fasttext.cc/docs/en/pretrained-vectors.html

Team	R_A	R_B	Classifier Approach	Text Features	Label (Post-) processing	Additional Data	Hierachical Model Categorization
EricssonResearch (Umaashankar and Shanmugam S, 2019)	1	2	Conv Seq2Seq	fastText	random oversampling	–	global
TwistBytes (Benites, 2019)	2	1	one-vs-all SVM	TF-IDF n-grams + char n-grams	LCA	–	local per parent
DFKI-SLT (Ostendorff et al., 2019)	3	4	Transformer (BERT)	BERT	–	Wikidata KG Embeddings	global
Averbis (Genc et al., 2019)	6	3	Global CNN	fastText	T Criterion	–	global
Raghavan (K et al., 2019)	4	–	one-vs-all SVM	TF-IDF bi-grams	label count classifier	–	–
Fosil-hsmw (Bellmann et al., 2019)	5	–	SVM chain	GloVe + fastText	–	Author Database from RH	–
HSHL (Rother and Rettberg, 2019)	7	5	Logistic Regression + Naïve Bayes	TF-IDF uni-grams	limit by threshold	–	local
COMTRAVO-DS (Batista and Lyra, 2019)	8	6	Local CNNs	fastText	–	–	local
HUIU (Andresen et al., 2019)	9	–	one-vs-all SVM	BOW n-grams	limit by threshold	–	–
Baseline	–	–	one-vs-all-SVM	BOW uni- & bi-grams	root path completion	–	local
Contender	–	–	capsule networks	fastText	root path completion	–	global

Table 4: Overview of submitted approaches.

in the same model, here traditional and neural network classifiers are employed.

A variety of solution approaches have been submitted, 4 teams used SVM classifiers, where Fosil-hsmw opted for an RBF kernel and TwistBytes, HUIU, and Raghavan used a linear kernel function. HSHL decided to use a combinded approach using Logistic Regression and Naïve Bayes, and 4 teams used neural network approaches, whereas 3 teams (EricssonResearch, COMTRAVO-DS, and Averbis) included convolutional layers in their architecture, and DFKI-SLT used an approach based on the transformer architecture (Vaswani et al., 2017), specifically BERT (Devlin et al., 2019). Whereas most teams used standard tokenization approaches such as spaCy, NLTK[16], scikit-learn, etc., Raghavan use a Byte-Pair-Encoding (BPE) approach for tokenization. With those more general pieces of words, team Raghavan can build a more general vocabulary with reduced size. As text-representation within the classifier architecture, 4 teams decided to used traditional sparse representations in form of TF-IDF feature vectors (TwistBytes, Raghavan, HSHL) based on token-, POS-, or character n-grams and varying n (mostly $n = \{1, 2\}$). Fosil-hsmw, EricssonResearch, DFKI-SLT, COMTRAVO-DS, and Averbis relied on pre-trained embeddings, whereas Fosil-hsmw and EricssonResearch also trained embeddings on the provided blurbs. fastText[17] (Bojanowski et al., 2017) was mostly selected as the embedding framework of choice due to its ability to account for sub-word information and thus better handling of out-of-vocabulary words.

Other (provided) metadata processing, e.g. the number of authors, age of a book, gender of the author(s), ISBN-part splitting, etc., has been employed by several teams: Fosil-hsmw, EricssonResearch, DFKI-SLT, HUIU, and Raghavan. Further, external data was used by 2 teams: DFKI-SLT used knowledge graph embeddings based on Wikidata[18], and Fosil-hsmw crawled the Random House website for additional author information to set up an author database and train task-specific embeddings.

Several teams studied the issue of label post-processing, i.e. the coherence of the hierarchy or more generally the number of labels to predict for a sample, by using several approaches: TwistBytes used a technique called LCA (Label Cardinality Adjustment; details can be found in their paper) for limiting the number of labels to predict, Averbis used a similar correction step as described in Section 5.1 named T-Criterion in order to correct non-connected child nodes, HSHL and HUIU used a threshold mechanism for the number of labels to predict (the threshold(s) were treated as a hyperparameter and optimized accordingly), and Raghavan used an independent prediction model for the number of labels. Motivated by the inherent imbalance of the sample size per

[16] http://www.nltk.org/

[17] https://fasttext.cc

[18] https://wikidata.org

label, EricssonResearch used random oversampling as a technique to balance the dataset.

6 Results and Discussion

6.1 Evaluation Metrics

Several metrics have been introduced to evaluate systems for hierarchical classification tasks, here, we use **micro-averaged** recall, precision, and F1-score and follow suggestions by Silla and Freitas (2011) and Sorower (2010). While macro-averaging, the respective scores are computed for each label individually and then averaged to produce a final single score; micro-averaged scores are computed globally for each metric over all instances. Thus, more frequent labels have a higher impact on the micro-averaged score, which essentially affects more general labels, since they appear more frequently in the dataset. Hence, we impose more importance on correct predictions on higher levels believing this yields to a more realistic scenario. (Silla and Freitas, 2011) suggest the use of micro-averaged scores for hierarchical classification tasks and even refer to them as hierarchical precision, recall, and F_1. However, these flat performance measures do not necessarily align with hierarchical ones, as shown in (Brucker et al., 2011), we thus additionally measure the hierarchical consistency score (HC) for Task B. This score measures the ratio of predictions made by the system that conform with the underlying label hierarchy, i.e. that all ancestors of a label are also assigned to the sample.

We further employ the exact match ratio or so-called *subset accuracy* (Acc) as described in (Sorower, 2010) because it captures how well labels are selected in relation to each other. In contrast to the F_1-score, which takes partially correct classifications into account, the subset accuracy is a very strict metric as there is no distinction between partially correct classification and completely incorrect classifications.

6.2 Quantitative evaluation

The extensive list of results during the test phase and the post-evaluation phase is shown in the appendix A and B. The following analysis is based only on the results of the best system submitted by each team during the test phase.

Task A: Scores of the best system submission from each team for Task A are listed in Table 6. The best performing system achieved a micro-F_1 score of 0.867 and was submitted by EricssonResearch[19]. Besides, this system has also achieved the highest subset accuracy with a significant margin to the second-highest score.

Further analysis of the scores for each top-level category shows that the system by EricssonResearch performed especially well on categories with the fewest samples in the dataset, i.e. *Architektur & Garten (Architecture and Gardening)* and *Künste (Arts)* as can be seen in Table 5. In contrast, our Baseline system performs the worst for these classes and lacks behind significantly to all submissions. For categories with a high number of examples such as *Literatur & Unterhaltung (Literature & Entertainment)*, all submitted systems perform equally, which indicates that the main challenge for Task A might be data sparsity. EriccsonResearch was the only team that explicitly addressed this issue by using random oversampling.

Task B: Results for Task B are listed in Table 7. Team TwistBytes submitted the system with the highest F_1 score of 0.6767. The subset accuracy score of 0.3791 of the system by EricssonResearch (2nd rank) is particularly interesting, outperforming all other teams by at least 11%. Regarding hierarchy conformity (HC), five out of six systems have a perfect score concerning the inherent category hierarchy (HC). Notably, the system submitted by DFKI-SLT has an almost perfect hierarchy consistency (HC) score although they do not directly encode any hierarchy information within their model. Again, the Baseline system was outperformed by a large margin, scoring lowest of all systems in terms of recall, but surprisingly also achieving the highest precision score.

The capsule network (contender) performs in the mid-range, while the only other global approach that outperforms the capsule network is by EricssonResearch.

Further analysis of F_1 scores on each hierarchy level shows a performance decline throughout all systems for categories on deeper, and thus sparser, levels (c.f. Figure 6 (a) and (b)).

[19] Note that team Raghavan submitted improved results in the post-evaluation phase that beat the best results of the test phase.

Team	*Literatur & Unterhaltung*	*Sachbuch*	*Kinderbuch & Jugendbuch*	*Ratgeber*	*Ganzheitliches Bewusstsein*	*Glaube & Ethik*	*Architektur & Garten*	*Künste*
EricssonResearch	**0.93**	0.75	**0.88**	**0.79**	0.78	0.75	**0.77**	**0.85**
twistbytes	0.92	0.76	0.87	**0.79**	**0.80**	**0.78**	0.71	0.74
DFKI-SLT	**0.93**	**0.78**	0.84	**0.79**	0.79	0.73	0.69	0.81
Raghavan	**0.93**	0.75	0.87	**0.79**	0.74	0.74	0.65	0.65
Fosil-hsmw	0.92	0.71	0.84	0.73	0.73	0.74	0.71	0.77
Averbis	0.92	0.71	0.82	0.73	0.77	0.74	0.56	0.68
HSHL	0.90	0.72	0.76	0.74	0.74	0.72	0.65	0.62
Comtravo-DS	0.90	0.71	0.78	0.76	0.74	0.73	0.65	0.67
HUIU	0.89	0.70	0.74	0.73	0.71	0.68	0.61	0.73
Contender	0.91	0.71	0.83	0.76	0.78	0.77	0.71	0.77
Baseline	0.90	0.68	0.69	0.72	0.69	0.63	0.34	0.45
#Samples in test set	2182 (49%)	650 (14%)	575 (13%)	536 (12%)	262 (6%)	183 (4%)	44 (1%)	38 (<1%)

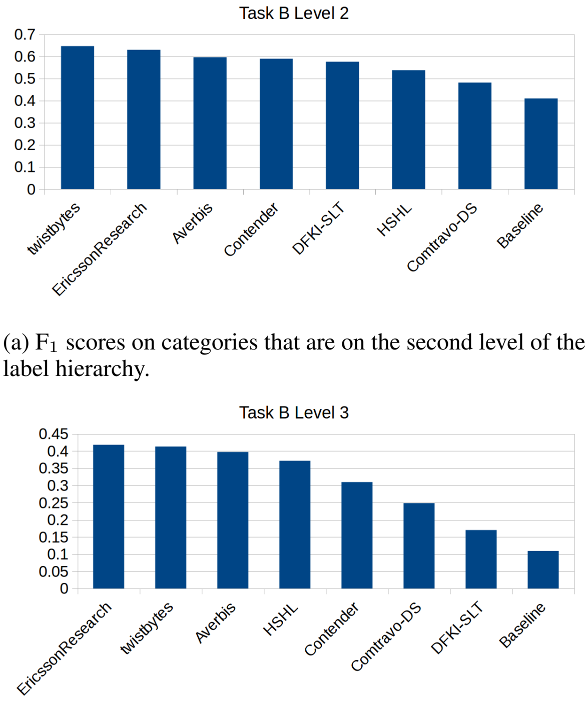
Table 5: F_1 scores for top-level categories for Task A.

Rank	best System by Team	Acc	Precision	Recall	F_1
1	EricssonResearch	**.84**	**.89**	.84	**.87**
2	TwistBytes	.79	.87	**.86**	.86
3	DFKI-SLT	.82	.88	.85	.86
4	Raghavan	.83	.88	.84	.86
5	Fosil-hsmw	.79	.84	.83	.84
6	Averbis	.79	.86	.81	.83
7	HSHL	.77	.82	.82	.82
8	Comtravo-DS	.72	.81	.83	.82
9	HUIU	.76	.81	.81	.81
	Contender	.74	.82	.85	.84
	Baseline	.71	.86	.75	.80

Table 6: Results for Task A of participating teams. Only the best performing system per team is listed. Scores are micro-averaged.

Rank	Model	Acc	Precision	Recall	F_1	HC
1	Twistbytes	.25	.71	**.65**	**.68**	1
2	EricssonResearch	**.38**	.74	.62	.67	1
3	Averbis	.27	.68	.61	.64	1
4	DFKI-SLT	.21	.78	.52	.62	.97
5	HSHL	.26	.72	.54	.62	1
6	Comstravo-DS	.19	.70	.53	.60	1
	Contender	.25	.76	.56	.64	1
	Baseline	.15	**.85**	.39	.53	1

Table 7: Results for Task B of all participating systems. Only the best performing system is listed. Illustrated scores are micro-averaged.



(a) F_1 scores on categories that are on the second level of the label hierarchy.

(b) F_1 scores on categories that are on the third level of the label hierarchy.

Figure 6: Performance report on different levels of the hierarchy.

7 Summary

We presented the summary report of the *GermEval-2019 Task 1: Hierarchical Classification of Blurbs* which included two sub-tasks: classification of categories of different granularities. As part of this shared task, participants were provided with a dataset consisting of blurbs including metadata in German of around 20K books. The shared task consisted of three phases: the first phase was designed to familiarize with the task and the data, the second phase provided the training data and a platform to compare the performance of submissions on the held-out validation set, and the third phase provided access to the validation data for additional training and disabled performance comparisons on the held-out test set for fairness purposes. System submissions cover a variety of approaches to deal with the category hierarchy: three systems (+ baseline) were designed using the local approach, either by learning one model (SVM or CNN) per parent node or per level. Four (+ contender) systems employed the global approach: three teams use CNNs and one uses transformer networks with a linear decoder on top. Most systems incorporated the hierarchy directly into their system or employed a post-processing step to adjust predictions. While some of the top-performing teams employed deep neural network architectures either for learning a representation of blurbs or for the classification task itself, well adjusted and fine-tuned traditional classifiers have shown competitive results.

Acknowledgments

We would like to thank the publisher group Random House for their permission to crawl their website and to make the dataset publicly available. We also congratulate all participants on their achievements and appreciate the diversity of the approaches. This work was partially supported by the Forum 4.0 project, funded by Hamburg's BWFG.

References

Rami Aly. 2018. Hierarchical writing genre classification with neural networks. B.Sc. Thesis, Universität Hamburg, Germany.

Rami Aly, Steffen Remus, and Chris Biemann. 2019. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

Melanie Andresen, Melitta Gillmann, Jowita Grala, Sarah Jablotschkin, Lea Röseler, Eleonore Schmitt, Lena Schnee, Katharina Straka, Michael Vauth, Sandra Kübler, and Heike Zinsmeister. 2019. The HUIU contribution to the GermEval 2019 shared task 1. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Simon Baker, Imran Ali, Ilona Silins, Sampo Pyysalo, Yufan Guo, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2017. Cancer hallmarks analytics tool (chat): a text mining approach to organize and evaluate scientific literature on cancer. *Bioinformatics*, 33(24):3973–3981.

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2015. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

David S. Batista and Matti Lyra. 2019. COMTRAVO-DS team at GermEval 2019 task 1 on hierarchical classification of blurbs. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Franz Bellmann, Lea Bunzel, Christoph Demus, Lisa Fellendorf, Olivia Graupner, Qiuyi Hu, Tamara Lange, Alica Stuhr, Jian Xi, Michael Spranger, and Dirk Labudde. 2019. Multi-label classification of blurbs with SVM classifier chains. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Fernando Benites. 2019. TwistBytes - hierarchical classification at GermEval 2019: walking the fine line (of recall and precision). In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5(1):135–146.

Florian Brucker, Fernando Benites, and Elena Sapozhnikova. 2011. An empirical comparison of flat and hierarchical performance measures for multi-label classification with hierarchy extraction. In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 579–589, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ricardo Cerri, Rodrigo C. Barros, and André C.P.L.F. de Carvalho. 2014. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80(1):39 – 56.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, US.

C.J. Fall, A. Törcsvári, P. Fiévet, and G. Karetka. 2004. Automated categorization of german-language patent documents. *Expert Systems with Applications*, 26(2):269 – 277.

Erdan Genc, Louay Abdelgawa, Viorel Morari, and Peter Kluegl. 2019. Convolutional neural networks for classification of German blurbs. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

José María Gómez Hidalgo, Guillermo Cajigas Bringas, Enrique Puertas Sánz, and Francisco Carrero García. 2006. Content based SMS spam filtering. In *Proceedings of the 2006 ACM Symposium on Document Engineering*, pages 107–114.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. Http://www.deeplearningbook.org.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Raghavan A K, Venkatesh Umaashankar, and Gautham Krishna Gudur. 2019. Label frequency transformation for multi-label multi-class text classification. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Kang-Min Kim, Yeachan Kim, Jungho Lee, Ji-Min Lee, and SangKeun Lee. 2019. From small-scale to large-scale text classification. In *The World Wide Web Conference*, WWW '19, pages 853–862, New York, NY, USA.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, Banff, Canada.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. HDLTex: Hierarchical deep learning for text classification. In *IEEE International Conference on Machine Learning and Applications*, pages 364–371, Cancún, Mexico.

Kristin Larsson, Simon Baker, Ilona Silins, Yufan Guo, Ulla Stenius, Anna Korhonen, and Marika Berglund. 2017. Text mining for improved exposure assessment. *PloS one*, 12(3):1–21.

David D. Lewis. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397.

Eneldo Loza Mencía and Johannes Fürnkranz. 2010. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic Processing of Legal Texts*, pages 192–215. Springer.

Malte Ostendorff, Peter Bourgonje, Maria Moritz, Julián Moreno-Schneider, and Georg Rehm. 2019. Enriching BERT with knowledge graph embeddings for document classification. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. 2015. Lshtc: A benchmark for large-scale text classification. *ArXiv:1503.08581*.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359.

Kristian Rother and Achim Rettberg. 2019. Logistic regression and naive bayes for hierarchical multi-label classification at GermEval 2019 - task 1. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Carlos N. Silla and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Mohammad S. Sorower. 2010. A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis, OR, USA.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 521–528, San Jose, CA, USA.

Rahim Taheri and Reza Javidan. 2017. Spam filtering in sms using recurrent neural networks. In *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pages 331–336, Shiraz, Iran.

Domonkos Tikk, György Biró, and Jae Dong Yang. 2005. *Experiment with a Hierarchical Text Categorization Method on WIPO Patent Collections*, pages 283–302. Boston, MA, USA.

Venkatesh Umaashankar and Girish Shanmugam S. 2019. Multi-label multi-class hierarchical classification using convolutional Seq2Seq. In *GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Ulli Waltinger, Alexander Mehler, Mathias Lösch, and Wolfram Horstmann. 2009. Hierarchical classification of OAI metadata using the DDC taxonomy. In *Advanced Language Technologies for Digital Libraries*, pages 29–40, Trento, Italy.

Liqiang Xiao, Honglun Zhang, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. MCapsNet: Capsule network for text with multi-task learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4565–4574, Brussels, Belgium.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*, pages 3110 – 3119, Brussels, Belgium.

A Full Results Task A

Rank	Team	System	F1	Precision	Recall	Support	Phase
1	Raghavan	SVM-BPEB	0.88	0.90	0.86	1.00	post-eval
2	EricssonResearch	CC_a_fconv_b_A6C1Y	0.87	0.89	0.84	0.98	post-eval
3	EricssonResearch	fconv_A6C1Y	0.87	0.89	0.84	1.00	test
4	twistbytes	baseline_lca	0.86	0.85	0.88	0.99	post-eval
5	twistbytes	sklearn_hier_threshold_and_roots_baseline_thresholding	0.86	0.86	0.86	0.96	test
6	DFKI-SLT	full	0.86	0.88	0.85	1.00	test
7	Raghavan	SVM-BPEB	0.86	0.88	0.84	1.00	test
8	twistbytes	baseline_0_25	0.86	0.82	0.90	1.00	post-eval
9	DFKI-SLT	text-only	0.86	0.87	0.84	1.00	test
10	EricssonResearch	fconv_F8V17	0.85	0.88	0.83	1.00	test
11	knowcup	DL_single_test	0.84	0.85	0.84	1.00	test
12	DFKI-SLT	full2	0.84	0.87	0.81	1.00	test
13	LT	Contender	0.84	0.82	0.85	1.00	test
14	fosil-hsmw	SVM_ECC	0.84	0.84	0.83	1.00	test
15	Averbis	BOHB_CNN	0.83	0.86	0.81	0.98	test
16	twistbytes	sklearn_hier_threshold	0.83	0.91	0.76	0.85	test
17	HSHL	LogisticRegression_NaiveBayes1	0.82	0.82	0.82	1.00	test
18	Comtravo-DS	local_clf_logit_cnn	0.82	0.81	0.83	0.94	test
19	HSHL	LogisticRegression_NaiveBayes2	0.82	0.82	0.81	1.00	test
20	HUIU	multi	0.81	0.81	0.81	1.00	test
21	LT	Baseline_wo_correction	0.80	0.86	0.75	0.88	test
21	LT	Baseline	0.80	0.86	0.75	0.88	test
22	Comtravo-DS	global_clf_cnn	0.78	0.78	0.78	0.99	test
23	EricssonResearch	fconv_4LYFP_7EKHC_WNG1A	0.66	0.68	0.64	1.00	test

B Full Results Task B

Rank	Team	System	F1	Precision	Recall	Support	Phase
1	twistbytes	sklearn_hier_threshold_and_roots_baseline_thresholding	0.68	0.71	0.65	0.98	test
1	twistbytes	sklearn_hier_threshold	0.68	0.71	0.65	0.98	test
2	EricssonResearch	fconv_A6C1Y	0.67	0.74	0.62	1.00	test
2	EricssonResearch	CC_a_fconv_b_A6C1Y	0.67	0.74	0.62	1.00	post-eval
3	EricssonResearch	fconv_F8V17	0.66	0.72	0.60	1.00	test
4	knowcup	DL_single_test	0.65	0.75	0.58	1.00	test
5	Averbis	BOHB_CNN	0.64	0.68	0.61	1.00	test
6	LT	Contender	0.64	0.75	0.56	1.00	test
7	DFKI-SLT	full	0.62	0.78	0.52	1.00	test
7	DFKI-SLT	full2	0.62	0.78	0.52	1.00	test
8	HSHL	LogisticRegression_NaiveBayes1	0.62	0.72	0.54	1.00	test
9	HSHL	LogisticRegression_NaiveBayes2	0.61	0.74	0.51	1.00	test
10	Comtravo-DS	local_clf_logit_cnn	0.60	0.70	0.53	0.94	test
11	DFKI-SLT	text-only	0.58	0.72	0.49	1.00	test
12	Comtravo-DS	global_clf_cnn	0.54	0.57	0.52	1.00	test
13	LT	Baseline	0.53	0.85	0.39	0.88	test
14	LT	Baseline_wo_correction	0.53	0.85	0.39	0.88	test
15	EricssonResearch	fconv_4LYFP_7EKHC_WNG1A	0.48	0.58	0.42	1.00	test
16	twistbytes	baseline_0_25	0.45	0.82	0.31	1.00	post-eval
17	twistbytes	baseline_lca	0.44	0.85	0.30	0.99	post-eval
18	twistbytes	thresholding	0.44	0.86	0.30	0.96	test
19	Raghavan	SVM-BPEB	0.39	0.70	0.27	1.00	post-eval
20	NoTeam	GRU_Attention_ensemble1	0.33	0.42	0.28	1.00	test