Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Datateknik
2021 | LIU-IDA/LITH-EX-A--21/038--SE

Automatic Categorization of News Articles With Contextualized Language Models

Automatisk kategorisering av nyhetsartiklar med kontextualiserade språkmodeller

Lukas Borggren

Supervisor : Ali Basirat
Examiner : Marco Kuhlmann

External supervisor : Hans Hjelm

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

This thesis investigates how pre-trained contextualized language models can be adapted for multi-label text classification of Swedish news articles. Various classifiers are built on pre-trained BERT and ELECTRA models, exploring global and local classifier approaches. Furthermore, the effects of domain specialization, using additional metadata features and model compression are investigated. Several hundred thousand news articles are gathered to create unlabeled and labeled datasets for pre-training and fine-tuning, respectively. The findings show that a local classifier approach is superior to a global classifier approach and that BERT outperforms ELECTRA significantly. Notably, a baseline classifier built on SVMs yields competitive performance. The effect of further in-domain pre-training varies; ELECTRA's performance improves while BERT's is largely unaffected. It is found that utilizing metadata features in combination with text representations improves performance. Both BERT and ELECTRA exhibit robustness to quantization and pruning, allowing model sizes to be cut in half without any performance loss.

Acknowledgments

First and foremost, I would like to thank Hans Hjelm and the Machine Learning Team at Bonnier News for providing valuable insights and guidance throughout the work on this thesis. I also want to direct a special thanks to David Hatschek at Bonnier News for his keen interest and engagement in the project. Finally, I want to thank my supervisor Ali Basirat and examiner Marco Kuhlmann at Linköping University for their feedback and encouragement during the thesis work.

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
ALBERT	A Lite BERT
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
DL	Deep Learning
ELECTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurately
FNR	False Negative Rate
FPR	False Positive Rate
GPT	Generative Pre-trained Transformer
HMTC	Hierarchical Multi-label Text Classification
HTrans	Hierarchical Transfer learning
IPTC	International Press Telecommunications Council
LCL	Local Classifier per Level
LCN	Local Classifier per Node
LCPN	Local Classifier per Parent Node
LM	Language Modeling
ML	Machine Learning
MLM	Masked Language Modeling
MLNP	Mandatory Leaf-Node Prediction
MSL	Maximum Sequence Length
NMLNP	Non Mandatory Leaf-Node Prediction
NLP	Natural Language Processing
NSP	Next Sentence Prediction
RoBERTa	a Robustly optimized BERT pre-training approach
SVM	Support Vector Machine
TC	Text Classification
TF-IDF	Term Frequency-Inverse Document Frequency
VM	Virtual Machine

1 Introduction

As many other parts of society, the news media industry has undergone – and continues to undergo – fundamental changes in the wake of rapid digitization. News dissemination and consumption have changed drastically over the last decades, causing traditional mediums such as printed newspapers to plummet in sales [41]. Instead, news are increasingly consumed online through websites and mobile apps in a plethora of formats, such as social media, news aggregators and podcasts – a trend that is further accelerated by the COVID-19 pandemic [43]. This digital transformation is tightly coupled with the conditions for online news publishing, and news media business models have continuously evolved as the streams of costs and revenues have shifted [41]. For news organizations, this poses both challenges and opportunities in terms of, for example, profitable advertising and subscription models. To remain competitive and enhance efficiency, newsrooms are increasingly employing algorithmic tools to gather, produce, organize and distribute content [21]. Today, these tools include artificial intelligence (AI) technologies, such as machine learning (ML) and natural language processing (NLP), and their relevance is only expected to grow [7].

Text classification (TC) is the procedure of assigning predefined labels to text, a widely applicable task that is fundamental in NLP [34]. Dating back to the 1960s, TC systems initially relied on a knowledge engineering approach, where classification is performed in accordance with a set of manually defined rules [56]. In the 1990s, this approach lost traction in favor of ML techniques, where an inductive process is employed to automatically build a classifier based on manually labeled text. A few early examples of methods used for this supervised learning of TC systems are support vector machines (SVMs) [30] and artificial neural networks (ANNs) [71]. Since the 2010s, much of TC research is based on deep learning (DL) methods, extending the capabilities of ANNs by creating larger and more complex models [34].

In the past few years, the NLP field has arguably experienced a paradigm shift through the introduction of the contextual DL architecture Transformer [68]. Transformer-based models such as Generative Pre-trained Transformer (GPT) [50] and Bidirectional Encoder Representations from Transformers (BERT) [20] quickly achieved state-of-the-art performance on numerous NLP tasks, including TC [3], upon their release. These models exemplify a novel approach of pre-training large language representation models on big text corpora

for general language understanding. Through fine-tuning, these general-purpose models can subsequently be applied to specific downstream tasks using relatively small amounts of data. Particularly BERT has spawned a multitude of derivatives further advancing the state of the art, such as A Robustly Optimized BERT Pre-training Approach (RoBERTa) [36], A Lite BERT (ALBERT) [32] and Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) [15], just to mention a few. Due to the vast textual resources, contextualized language models are predominately pre-trained on English corpora, but models for lower resource languages such as Swedish are starting to emerge [39].

This thesis investigates how pre-trained contextualized language models can be used to accurately categorize Swedish news articles. It is conducted in collaboration with Bonnier News, a Swedish news media group composed of the newspapers and related businesses owned by the Bonnier Group. With more than five million daily readers distributed over 100 newspapers, magazines and websites, it is one of Scandinavia's largest media groups.

1.1 Motivation

In this thesis, the classification task is characterized by the categories being ordered in a hierarchical structure, and each news article may be labeled with multiple categories. This type of TC problem is generally referred to as hierarchical multi-label text classification (HMTC). As Transformer-based models such as BERT are very recent, they have not yet been widely adapted for all types of NLP tasks, including HMTC. This thesis adds to the limited body of work that applies contextualized language models to HMTC. Furthermore, Swedish pre-trained language models are even more recent and have not been extensively studied. Accordingly, this thesis contributes to the small collection of research utilizing contextualized language models for Swedish NLP tasks.

A recent initiative at Bonnier News has had the purpose of producing a unified content system across all company brands. This includes a new taxonomy that will be used internally for categorizing news articles based on the global standard Media Topics[1], originally developed by the International Press Telecommunications Council (IPTC). Currently, there is an interest in implementing a system for automatic categorization of news articles according to this new taxonomy. There would be three main benefits with such a system for Bonnier News. Firstly, it would make it feasible to perform large-scale data backfills by re-categorizing older articles, which would create a unified structure in the data warehouse. Secondly, algorithmic categorization would facilitate a more consistent usage of the taxonomy across the entire corporate group, ideally free from human errors and biases. Thirdly, the workload of the journalists, who are currently tagging their articles manually, would be reduced.

1.2 Aim

The purpose of this thesis is to investigate and evaluate approaches to applying pre-trained contextualized language models to news article classification. Specifically, the main focus is on adapting Swedish versions of pre-trained BERT and ELECTRA models for HMTC and comparing them to a non-neural baseline. By utilizing different fine-tuning strategies and model architecture enhancements, the goal is ultimately to build a classification pipeline that can reliably categorize Swedish news articles. Moreover, the aim is for this pipeline's memory consumption and inference time to be non-prohibitive in practical use cases.

[1] https://iptc.org/standards/media-topics/

1.3 Research Questions

This thesis aims to answer the following research questions. Classification performance is evaluated in terms of exact match ratio and micro-averaged precision, recall and F_1 score.

1. *What performance can be achieved on news article classification using pre-trained contextualized language models, employing a global and a local classifier approach, respectively?*

   There are various methods to handle a hierarchical structuring of class labels. Two approaches are to use a global classifier, where a single classifier considers the entire flattened class hierarchy, or to use local classifiers, where multiple classifiers are designated to delimited parts of the class hierarchy.

2. *How does further in-domain pre-training affect performance?*

   Fine-tuning of a pre-trained model for TC is a supervised learning task that requires labeled data. However, it is possible to specialize a pre-trained model for a specific domain prior to fine-tuning. This involves resuming unsupervised pre-training using unlabeled data from a similar distribution as the labeled task data.

3. *How can additional metadata features be used to improve performance?*

   TC models generally rely on a document representation derived solely from textual features extracted from the document itself. Nonetheless, documents may have associated metadata that can function as highly discriminative features for classification.

4. *What effect does model quantization and pruning have on performance?*

   Contextualized language models are notoriously large and resource intensive. Quantization and pruning are model compression techniques that can be utilized to reduce inference and memory costs, while retaining adequate classification performance.

1.4 Delimitations

This thesis only considers Swedish text from the news media domain for training and evaluation of the proposed methods. The pre-trained models employed are all monolingual and have been trained exclusively on Swedish corpora. Additionally, the effects of compression techniques on compute and memory resource consumption are not investigated.

2 Theory

To provide a theoretical framework for this thesis, the following chapter is devoted to presenting background literature and previous research related to the thesis' subject area.

2.1 Text Classification

Text classification (TC), also called text categorization or document classification, can be defined as the problem of assigning a boolean value T or F to each pair $(d_i, c_j) \in D \times C$, where D is a domain of documents and $C = \{c_1, ..., c_{|C|}\}$ is a set of predefined classes [56]. If the value assigned to (d_i, c_j) is T, the document d_i should be labeled with class c_j, and if the assigned value is F, d_i should not be labeled with c_j. More formally, the problem is to create a classifier $\hat{\Phi} : D \times C \rightarrow \{T, F\}$ that approximates the unknown target function $\Phi : D \times C \rightarrow \{T, F\}$ such that $\hat{\Phi}$ and Φ coincide optimally, according to some evaluation criterion. The most basic problem formulation is binary TC: a single-label classification problem where $|C| = 2$ and each $d_i \in D$ is assigned to either c_1 or its complement $\bar{c_1} = c_2$.

2.1.1 Multi-Label Classification

Multi-label TC is the problem where the number of classes $|C| > 2$ and every document d_i is labeled with a set of classes $C_{d_i} \subseteq C$, where $|C_{d_i}| \geqslant 1$. Consequently, a document can be assigned to multiple classes, which should be distinguished from multi-class problems where a document is assigned to exactly one of multiple classes, that is $|C_{d_i}| = 1$. There are two general approaches to multi-label classification: problem transformation and algorithm adaption [66]. Problem transformation refers to methods modifying the data to partition the problem into multiple single-label classification problems, while algorithm adaption are methods for altering classification models to directly handle multi-labeled data.

For multi-class classification, the output from a classifier is generally a probability distribution over all the class labels. Then, the label with the highest probability is typically the resulting prediction. For multi-label classification however, a single prediction may consist of several labels. Consequently, some classification threshold is needed to determine at what probability individual label predictions should be cut off. Commonly, classification thresholds are simply set to 0.5, but this may be unsuitable for problems with imbalanced class

distributions [80]. This is because classifiers generally minimize an average loss that is highly influenced by majority classes, causing the predictions to be skewed towards them. There are several strategies for optimizing the classification thresholds for TC problems. One of the most common strategies is SCut, a metric-agnostic algorithm that optimizes classification performance individually for each class. SCut involves tuning a separate threshold for each class to optimize some metric on the validation set [74]. Subsequently, the per-class thresholds are fixed when applying the classifier to new documents in the test set. However, the applicability of SCut is problem-specific, as it risks overfitting the thresholds on the validation data for certain datasets.

2.2 Hierarchical Classification

In hierarchical classification problems, the classes C are organized in a hierarchical taxonomy, which is most commonly structured as a tree [59]. If such a hierarchy has n levels, the classes can be divided into n disjoint sets, one for each level, where sets for adjacent levels have parent-child relations. That is, $C = \{C_1, ..., C_n\}$ and C_i contains the parents to the classes in C_{i+1}. In a sense, all hierarchical TC problems can be viewed as multi-label TC problems, as long as at least one class from at least one hierarchy level can be assigned to a document. However, a more stringent delimitation is to explicitly define hierarchical multi-label TC (HMTC) as problems where multiple classes from each hierarchy level may be assigned to a document. In general, every document d_i is labeled with a set of sets of classes $C_{d_i} = \{C_{d_i,1}, ..., C_{d_i,n}\} \subseteq C$, where $C_{d_i,k} \subseteq C_k$ and $|C_{d_i,k}| \geqslant 0$ for every hierarchy level k. In HMTC, it is commonly assumed that if a document has a ground truth label c_j, the document also has ground truth labels for all ancestors of the c_j node in the tree, forming a path up to the root node. Hierarchical classification problems can be distinguished by how deep in the hierarchy predictions must be made. Always assigning one or more leaf-node classes from C_n is referred to as mandatory leaf-node prediction (MLNP). Conversely, terminating classification at possibly any hierarchy level is referred to as non-mandatory leaf-node prediction (NMLNP). There are multiple approaches to address – or not address – the hierarchical structuring of classes. Two of the most widely explored ones are the global classifier approach and local classifier approaches.

2.2.1 Global Classifier Approach

The global classifier, also called "big-bang", approach refers to learning a global model for all classes in the hierarchy. This approach does not intrinsically exploit the class hierarchy, but has the benefit of only requiring a single classifier. A global classifier is typically a complex model that is trained on multi-label classification across the entire flattened hierarchy simultaneously. Consequently, a global classifier can potentially assign classes from every level to an example during a single inference run. Since a global classifier does not take the hierarchy into account, it is prone to produce class-membership inconsistencies. This occurs when a class is assigned to an example, but one or more of the class's ancestors are not, violating the hierarchical structure. To manage such inconsistencies, classification can be succeeded by a separate post-processing step that enforces the hierarchical constraints.

2.2.2 Local Classifier Approaches

Local classifier approaches utilize multiple models, each one designated to a subset of the class hierarchy. These approaches make use of local information from the hierarchy to specialize classifiers by reducing the class output space. Three standard techniques for utilizing this local information is to have a local classifier per level (LCL), a local classifier per parent node (LCPN) and a local classifier per node (LCN). While these techniques make increasingly more use of local information, a drawback is that they also require an increasing number of

classifiers. In Figure 2.1, an example of the LCL approach is shown, where each dashed rectangle represents a local classifier.

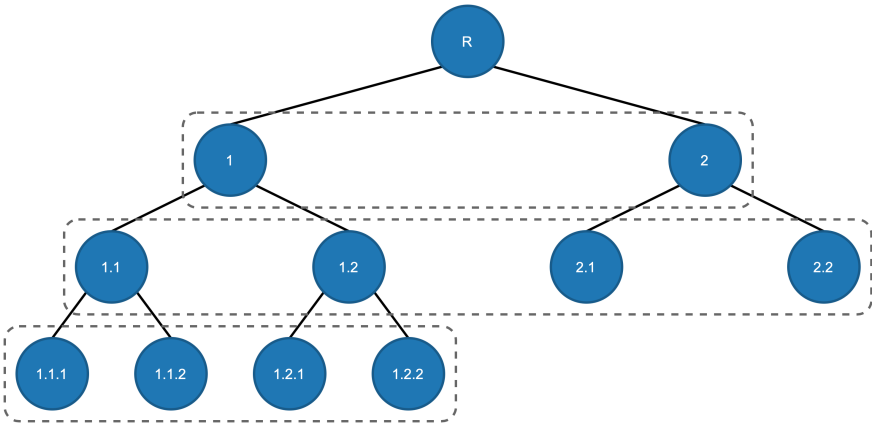


Figure 2.1: Local classifier per level approach

Even though local classification approaches exploit local information to make individual predictions, the process of combining the local classifiers' predictions is prone to produce class-membership inconsistencies. To avoid violations of the hierarchical constraints, the class-prediction top-down approach may be used. It refers to utilizing, for each level in the hierarchy, the predicted classes from the previous level to make decisions about the classification output at the current level. Specifically, only the children of classes predicted at the previous level are considered as possible class-label candidates. When using the top-down approach in conjunction with NMLNP, some stopping criterion must be used to control the depth of classification. A straight-forward method for achieving this is to have a threshold for each class node; then, further classification down a path in the hierarchy is terminated when a classifier outputs a probability lower than the threshold for a node in the path. However, this thresholding might cause classification errors to propagate downwards in the hierarchy, potentially blocking more specific classes from being assigned. There are different strategies for reducing this blocking problem, but their effectiveness is debatable or they require extensive additional learning processes [58]. For these reasons, such blocking reduction strategies will not be further considered in this thesis.

2.2.3 Hierarchical Transfer Learning

Transfer learning is the process of trying to improve the performance on a new task by utilizing previously learned knowledge from a related task [79]. This transfer of knowledge can take place across different domains and tasks, or both of them simultaneously. For ANNs, a common method is to employ parameter-based transfer learning, which refers to reusing the parameters of an already trained model to facilitate the learning of a new model. This approach is motivated by the fact that the parameters of a model reflect the inherent knowledge of the model. Hierarchical transfer learning (HTrans) is a strategy of training local classifiers for HMTC problems [6]. In HTrans, local classifiers are recursively trained in a top-down fashion by initializing each new child category classifier with the parameters of its parent category classifier and subsequently fine-tune the new classifier. By providing local classifiers with better starting points, classification performance can be improved, especially for lower hierarchy levels where classes are generally sparser.

2.3 Transformer

The Transformer is a DL architecture for sequence transduction that is the foundation for many of the advances in NLP in recent years. When released, it differed from many contemporary model architectures in that it discards any recurrence or convolutions and instead relies solely on attention mechanisms.

2.3.1 Attention

Attention is a DL mechanism used for determining some notion of relevance across the positions in a sequence [24]. It can be described as a function that maps a query and a set of key-value pairs to an output. Typically, queries, keys and values are all vectors that represent sequence positions. The output is a distribution over all values, where each value is weighted through some compatibility function between the corresponding key and the query. This distribution is the attention for the sequence position represented by the query, where the influence of the other sequence position values is determined by their relevance.

In NLP, attention was first introduced as a part of an encoder-decoder architecture in the context of neural machine translation. An encoder-decoder architecture consists of two main components: an encoder network followed by a decoder network. From an input sequence $x = (x_1, ..., x_n)$, the encoder creates hidden representations $z = (z_1, ..., z_n)$, which are subsequently passed to the decoder to construct an output sequence $y = (y_1, ..., y_m)$ [14]. In the work introducing attention in NLP, the encoder and decoder are both recurrent neural networks, x is the original sentence and y is the translated sentence. An attention mechanism is utilized in the decoder for mapping z to y, thus creating an alignment between the pair of sequences [5]. Intuitively, this induces the decoder with a level of context awareness, as it is able to attend to the most relevant parts of the input sequence for each translation step. This idea was subsequently developed through the introduction of self-attention, also called intra-attention [13]. Initially, it involved employing an attention mechanism over a single sequence in an encoder, creating undirected relations between relevant positions within the sequence itself. The result is a sequence representation encoded with information about lexical relationships between positions.

Scaled Dot-Product and Multi-Head Attention

The specific attention function employed in the Transformer is called scaled dot-product attention [68], shown in Equation 2.1. The queries and keys are vectors of dimensions d_k and the values are vectors of dimension d_v. In practice, attention is computed for a set of queries simultaneously with the queries, keys and values packed in the matrices Q, K and V respectively.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Furthermore, attention is calculated across multiple representation subspaces concurrently in a process called multi-head attention, shown in Equation 2.2. The queries, keys and values are all transformed h times with learned linear projections W_i^Q, W_i^K and W_i^V. The attention function is then computed in parallel for each of the projected versions, yielding h d_v-dimensional outputs. Lastly, the outputs are concatenated and linearly projected with W^O to create the final output attention.

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, ..., head_h)W^O \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.2}$$

2.3.2 Model Architecture

The Transformer is an encoder-decoder architecture, where the encoder and decoder are stacks of N identical layers, respectively, interconnected in sequence [68]. Figure 2.2 shows the overall model architecture. The encoder layers consist of two sub-layers: a multi-head self-attention mechanism and a fully connected feed-forward network. Around each sub-layer, there is a residual connection followed by layer normalization. The decoder layers consists of three sub-layers. Two of them are similar to the encoder sub-layers, but with a slightly modified multi-head attention that masks parts of the input sequence. The third sub-layer performs multi-head attention over the output representation from the encoder. Additionally, learned linear transformations and a softmax function are applied to the decoder output to produce the final output predictions. Both the encoder and decoder stacks use learned embeddings and positional encodings to construct representations from the input sequences.

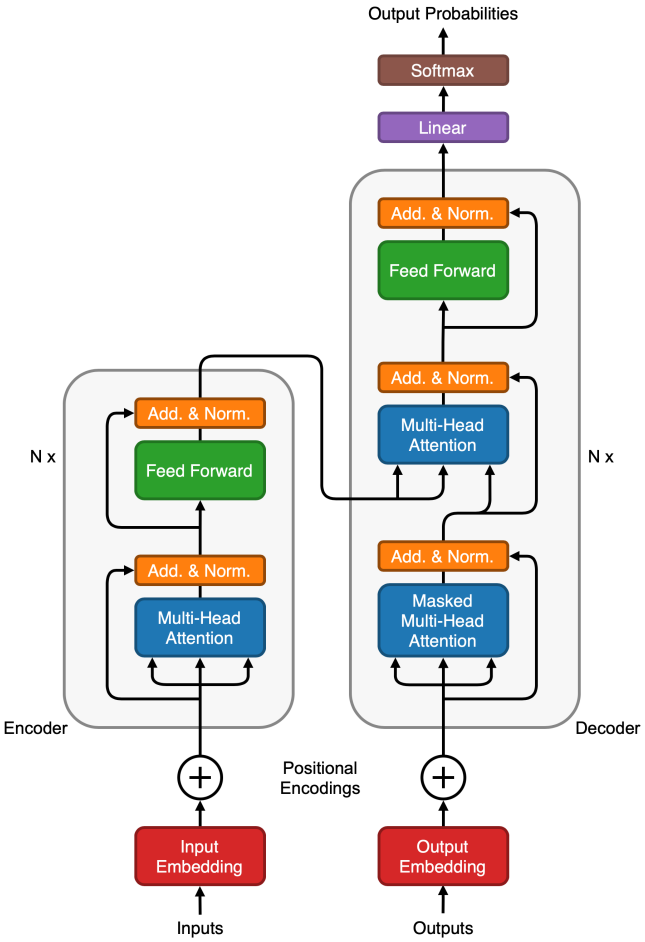


Figure 2.2: Transformer model architecture

Employing the Transformer for sequence-to-sequence modelling, a complete sequence x is inputted to the encoder, where self-attention is performed, outputting the representations z. Subsequently, the decoder produces the predictions y stepwisely at every output sequence position i, by performing attention over z and the produced outputs so far, $y_{i-1} = (y_1, ..., y_{i-1})$. Arguably the most prominent advantage of the Transformer over its precedents is that self-attention enables the model to learn dependencies between positions independent of their relative distances within the sequence.

2.4 Contextualized Language Models

In the last couple of years, Transformers have been extensively adapted for creating pre-trained contextualized language models. Language modeling (LM) is an NLP task where a model learns to predict the next word in a sequence based on the previous words in the sequence. When using Transformers in LM, language models are trained to utilize the contextual relations between words to make predictions, which has rapidly improved performance on the task [40]. Additionally, language models have proven to be highly generalizable when used as a basis for transfer learning in NLP. This has spawned a dominating paradigm in the NLP field, where general-purpose contextualized language models are pre-trained on large amounts of data and later repurposed for downstream tasks through fine-tuning. When fine-tuning such a pre-trained model on an NLP task, training is faster and requires less data, compared to training a comparable task-specific DL model from scratch. Since its inception, this paradigm has continuously pushed the state-of-the-art results on numerous tasks, benchmarks and datasets.

2.4.1 BERT

One of the first pre-trained contextualized language models, and arguably the most influential one to date, is Bidirectional Encoder Representations from Transformers (BERT). The idea of BERT is to pre-train deep bidirectional text representations from unlabeled data, that can subsequently be fine-tuned for downstream tasks relatively inexpensively and with minimal modifications of the model architecture [20]. BERT's architecture mainly consists of multiple layers of Transformer encoder blocks. Furthermore, BERT employs an input representation composed of the sum of learned token, segment and position embeddings. The token embeddings are learned based on a WordPiece vocabulary [73] created from the pre-training data. A special classification token [CLS] is added in front of every tokenized sequence, whose corresponding final hidden state is used as an aggregate sequence representation for classification tasks. Some NLP tasks require a pair of sentences as input, which BERT handles by concatenating two sentences in a single sequence and delimiting them by a special separation token [SEP]. In addition, separate segmentation embeddings are learned to differentiate sentences in a pair. The position embeddings are learned to represent absolute positions in the input sequence.

BERT is pre-trained on two unsupervised tasks: masked LM (MLM) and next sentence prediction (NSP). In MLM, a proportion of the input tokens are masked and the model's task is to predict these masked tokens. During training, 15% of the input tokens are randomly selected for masking and of those, 80% are replaced with a special [MASK] token, 10% are replaced with a random token and 10% are left unchanged. The reason for not replacing all 15% of the tokens with the [MASK] token is to mitigate possible effects of the special token not appearing in subsequent fine-tuning data. In NSP, two sentences are paired together and the model's task is to predict whether or not the second sentence follows the first one in the original text. During training, the second sentence actually follows the first one in 50% of the examples and for the remaining examples, the second sentence is randomly sampled from the corpus. BERT is pre-trained jointly on MLM and NSP by summing their losses, as shown in Equation 2.3. The combined loss is minimized with regard to the model parameters θ over a large unlabeled text corpus χ, comprising of tokenized input sequences $x = [x_1, x_2, ..., x_n]$ where $n \leqslant 512$.

$$\min_{\theta} \sum_{x \in \chi} \mathcal{L}_{MLM}(\boldsymbol{x}, \theta) + \mathcal{L}_{NSP}(\boldsymbol{x}, \theta) \tag{2.3}$$

Upon its release, BERT achieved state-of-the-art results for a wide range of NLP tasks. However, it has since then been superseded by various approaches that have identified and reme-

died shortcomings in the original pre-training formulation, such as XLNet [75] and A Robustly Optimized BERT (RoBERTa) [36]. RoBERTa, for instance, retains the overall architecture of BERT but improves the pre-training procedure by, among other things, employing dynamic masking during MLM and dispensing with the NSP objective altogether.

2.4.2 ELECTRA

Similar to RoBERTa, Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) is an approach to enhance BERT's pre-training. ELECTRA utilizes two models for its pre-training objective, a generator and a discriminator [15]. Figure 2.3 displays an overview of the pre-training setup. Both models have the same general architecture as BERT, but the generator is typically a quarter to half the size of the discriminator. The generator is trained to perform MLM, where 15% of the input tokens are randomly replaced with [MASK] tokens and the model's task is to predict the masked tokens. Concurrently, the discriminator is trained on a task called replaced token detection. Using the generator output as input, the discriminator's task is to predict, for every position in the input sequence, whether or not the token has been replaced by the generator.

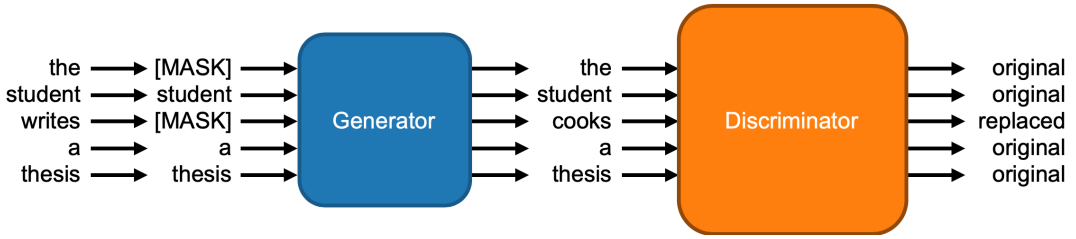


Figure 2.3: ELECTRA pre-training

The two models are trained jointly by combining their losses, as shown in Equation 2.4. The combined loss is minimized over both the generator's and the discriminator's parameters θ_G and θ_D. The weight λ for the discriminator loss $\mathcal{L}_{Disc}$ is empirically set to 50. After pre-training, the generator is disposed of and the discriminator can be fine-tuned on downstream tasks in the same way as other pre-trained language models.

$$\min_{\theta_G,\theta_D} \sum_{x \in \chi} \mathcal{L}_{MLM}(x, \theta_G) + \lambda \mathcal{L}_{Disc}(x, \theta_D) \tag{2.4}$$

Compared to preceding models, ELECTRA's pre-training objective is more efficient [15]. This allows for comparatively small ELECTRA models to perform on par with substantially larger versions of similar models. Additionally, when ELECTRA is scaled up and trained for longer, it outperforms previous state-of-the-art models, including BERT, RoBERTa and XLNet.

2.4.3 Swedish Language Models

Any ML model is limited by the data that it has been trained on. For pre-trained language models, this implies that their application is restricted to downstream tasks in the same language as the pre-training corpus. A possible means for increasing applicability is to create multilingual contextualized language models that are pre-trained on corpora containing text in multiple languages [48]. Such multilingual models can perform reasonably well, but have consistently been outperformed by monolingual models tailored to a single language [42, 70, 69]. Currently, there are two publicly available Swedish pre-trained BERT models: KB-BERT [39] developed by KBLab at Kungliga biblioteket (the National Library of Sweden) and SweBERT[1] created by Arbetsfömedlingen (the Swedish Public Employment Service). A third

[1] https://github.com/af-ai-center/SweBERT

Swedish BERT also exists, developed by the company BotXO, but there is limited documentation on how it performs and has been trained.[2] KB-BERT generally outperforms SweBERT and a multilingual BERT on Swedish TC tasks by a substantial margin [28].

Text Type	BERT	ELECTRA
Swedish Wikipedia	161 MB	161 MB
Government Text	834 MB	5,000 MB
Legal E-Deposits	400 MB	400 MB
Social Media	163 MB	5,000 MB
Newspapers	16,783 MB	90,000+ MB
Books	0 MB	2,000 MB
Total	18,341 MB	100,000+ MB

Table 2.1: Corpora dispositions for Swedish pre-trained BERT and ELECTRA

Apart from KB-BERT, KBLab has also created Swedish pre-trained ELECTRA models.[3] These have been pre-trained on a larger corpus than KB-BERT; the dispositions of both corpora are shown in Table 2.1. There is no formal documentation published about the pre-training process of the Swedish ELECTRAs and the disposition of the training corpus is retrieved from a recorded webinar segment[4] held by the Director of KBLab.

2.4.4 Domain Specialization

Contextualized language models are typically pre-trained on generic corpora to generalize well across domains. However, performance in individual domains can be improved by performing pre-training on domain-specific corpora. This may be achieved by either pre-training a domain-specific model from scratch or performing further in-domain pre-training of an already pre-trained model. Applying these approaches to BERT, the performance on downstream tasks in specialized domains has been shown to improve significantly [33, 8, 12].

2.5 Text Classification Models

Traditionally, a TC pipeline includes steps for preprocessing, feature extraction, classification and evaluation [34]. Preprocessing is the preparatory procedure of cleaning text data, which may involve removal of unwanted words or characters. From this cleaned data, features are extracted that aim to reflect some descriptive properties of the text document. Subsequently, a classifier is created that predicts class labels based on the features. Lastly, the classifier is evaluated by comparing its predictions to a ground truth according to some metric. In DL, feature extraction and classification is generally performed jointly, as the model learns a set of non-linear transformations to map representations of the preprocessed text directly to output predictions. Fine-tuning pre-trained language models is an example of the latter.

2.5.1 Fine-Tuned BERTs

Already in the original paper, a method for adapting BERT for TC was proposed, which involves adding a classification head on top of the pre-trained model [20]. Specifically, a fully connected linear layer with weights $W \in \mathbb{R}^{K \times H}$ is added after the final hidden representations $C \in \mathbb{R}^{H}$ corresponding to the [CLS] token, where K is the number of classes

[2] https://github.com/botxo/nordic_bert
[3] https://huggingface.co/KB
[4] https://www.youtube.com/watch?v=9EkjfNJzy6s

and H is the hidden size of the pre-trained model. This classification layer has a softmax activation function that outputs a probability distribution over the class labels. The model is fine-tuned end-to-end by minimizing the cross-entropy loss between the output estimation $softmax(CW^T) = \hat{y}$ and ground truth label y, that is $\mathcal{L} = -y \log(\hat{y})$ for a single example. This type of fine-tuning approach has become somewhat canonical when adapting BERT for TC problems, including multi-label TC where the softmax activation and cross-entropy loss are swapped out for a sigmoid activation and binary cross-entropy loss [3, 64, 26, 39, 33, 8].

Some works have investigated how the canonical fine-tuning process of BERT for TC can be improved [64]. Among other things, it has been shown that further in-domain pre-training of BERT prior to fine-tuning improves the performance on TC. A limitation of BERT and many of its successors is that the maximum sequence length (MSL) of the input is fixed, typically to 512 tokens, necessitating truncation of texts that exceed this limit. For TC, a strategy of concatenating the beginning and end of long texts has proven to be preferable [64]. There exist more sophisticated methods for handling long input sequences [78, 9, 23], but these utilize model architectures that diverge significantly from the original BERT. The result is that the supply of pre-trained models of this type is scarce and there exist, for instance, no Swedish versions. It is also possible to improve the performance of BERT on TC by utilizing additional, non-textual metadata features [44]. This has been achieved by concatenating the aggregate hidden sequence representation with vectorized metadata, whereupon it is fed through a pair of fully connected layers with ReLU activations, followed by a final classification layer. Additionally, there are a number of more complex adaptions of BERT-based models for TC that combines the pre-trained models with other techniques [76, 49, 11, 38, 17]. However, these approaches are beyond the scope of this thesis and will not be further considered.

2.5.2 Non-Neural Models

As mentioned, traditional TC pipelines rely on separate steps for creating feature representations and performing classification. One of the strongest and most effective of these non-neural approaches is to use support vector machines (SVMs) in combination with term frequency-inverse document frequency (TF-IDF) representations [18, 19].

TF-IDF

One of the most basic document representation models is bag of words. Given a vocabulary of terms, a document is represented by a vocabulary-length vector whose elements are weighted according to the number of occurrences of the corresponding term in the document [54]. This weighting scheme is called term frequency and can be denoted as $tf_{t,d}$ for a term t and a document d. TF-IDF representations extend this idea by employing a more elaborate weighting scheme that includes inverse document frequency. Document frequency is defined as the number of documents in a collection that a term occurs in. Following [54], inverse document frequency for a term t is defined in Equation 2.5, where N is the number of documents in the collection and df_t is the document frequency of t. The composite TF-IDF weighting for a term t and document d is then defined in Equation 2.6.

$$idf_t = log \frac{N}{df_t} \tag{2.5}$$

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t \tag{2.6}$$

SVM

SVM is an ML method that functions like a large-margin classifier. Given some binary-labeled training data, the objective of an SVM classifier is to find a decision boundary that best separates the two classes in a vector space. This is done by maximizing the distance, or margin,

from the closest data points between classes, called support vectors, to the decision hyperplane. Training an SVM, a weight vector $\vec{w}$ that is orthogonal to the decision hyperplane is learned, together with an intercept term b. Following [54], the SVM classifier is defined in Equation 2.7, where $\vec{x}$ is a data point and the class labels are -1 and $+1$.

$$f(\vec{x}) = sign(\vec{w}^T\vec{x} + b) \tag{2.7}$$

SVMs are flexible classifiers in that they are based on kernel functions, which are used to control the properties of the decision boundary. For linearly separable problems, a basic linear kernel can be used, and for non-linear decision boundaries, the kernel can be, for example, a radial basis function. Furthermore, SVMs can be extended to work as soft-margin classifiers to allow for some misclassification during training in favor of greater generalizability. The trade-off between these two aspects is controlled by the regularization parameter C. Since SVMs are inherently binary, there exist various strategies for adapting them to problems with multiple classes. The most common technique is the one-versus-rest strategy, where a separate SVM is trained for each class. SVMs are particularly suited for TC because the problems are often characterized by a high-dimensional input space, few irrelevant features, sparse feature vectors and linear separability – all of which SVMs are good at handling [30]. It is common to use rather simple linear SVMs for TC problems, partially because the choice of kernel function does not have a significant impact on classification performance [63].

2.6 Model Compression Methods

Transformer-based language models are large: the base-size version of BERT consists of 110 million parameters [20]. As a result, they are resource intensive, both in terms of memory consumption and computational overhead, and by extension in terms of energy costs. A line of research that addresses these disadvantages is model compression, which has lately been extensively explored for BERT and its derivatives. Three common compression methods include quantization, pruning and knowledge distillation [25]. The latter involves training a smaller student model using the outputs from a larger pre-trained teacher model. Knowledge distillation can reduce model sizes significantly while retaining much of the performance of the original model on various tasks [65, 53, 29], including TC [3, 2]. Model compression is a trade-off between performance and model size, and two prevalent research directions are to either compress with minimal performance degradation or to compress maximally. This thesis focuses on the former, for which quantization, pruning and knowledge distillation to relatively large student models are effective [25]. However, knowledge distillation adds a considerable computational overhead when student models are relatively large [53, 67] and causes significant performance loss when student models are relatively small [65, 25, 2]. For these reasons, knowledge distillation will not be further considered in this thesis.

2.6.1 Quantization

Quantization, or data quantization, refers to reducing the number of bits used to represent model parameters [25]. In DL, weights are commonly represented by 32-bits floating point (FP32) numbers. By approximating these values with lower resolution, for example with 16-bit floating points (FP16) or 8-bit integers (INT8), the memory footprint and inference time of a model can be reduced significantly, while the precision of its numerical calculations is lowered [35, 31]. Generally, quantization can be applied to all weights in BERT-like models [25], but it can be favorable to create mixed-precision models, where certain layers that are sensitive to quantization are kept at full precision [57]. Quantization can be applied post-training, which adds no computational overhead [35]. Alternatively, quantization-aware training can be used to reduce potential performance loss, but with the cost of performing additional training steps to adjust the quantized parameters [57, 77].

Quantization methods differ in what quantization scheme they employ, that is, how they map values from full resolution to the target bit-resolution. Commonly, these schemes are based on constructing some scaling factor that is multiplied with the original values to compute the new bit representation. For example, floats can be uniformly quantized to unsigned integers of k bit-precision in the range $\{0, ..., 2^k - 1\}$. Following [35], the weights W of a model can then be quantized as shown in Equation 2.8.

$$\begin{aligned} W' &= Clamp(W, q_0, q_{2^k-1}) \\ W^I &= \left\lfloor \frac{W' - q_0}{\Delta} \right\rceil, \qquad \Delta = \frac{q_{2^k-1} - q_0}{2^k - 1} \\ Quantize(W) &= \Delta W^I + q_0 \end{aligned} \tag{2.8}$$

In the formulas, $[q_0, q_{2^k-1}]$ denotes the quantization range, where q_0 is commonly referred to as offset; $Clamp()$ is a function clamping all elements to the quantization range; Δ is the distance between two adjacent quantized points and inverted, it is the scaling factor; W^I is a set of integer indices; and $\lfloor\rceil$ is the rounding operator. The quantization range, and thereby the scaling factor, can be determined both statically and dynamically. That is, it may be computed either prior to or during training, or alternatively, inference. Note that $W^I \in \{0, ..., 2^k - 1\}$ are the actual integer representations of the weights and $Quantize()$ subsequently maps these to the quantization range using the scaling factor and offset.

2.6.2 Pruning

Pruning, or weight pruning, is a compression method that locates and removes lesser important parameters or redundancies in models [25]. There are two main approaches to pruning: structured pruning and elementwise pruning. Structured pruning focuses on reducing and simplifying architectural components of BERT-based models. For example, the number of attention heads or Transformer blocks may be reduced. Elementwise pruning instead targets pruning of individual weights by identifying the set of least important weights of a model. Pruning is then performed by zeroing out the identified weights, which effectively translates to removing connections between neurons. The notion of importance can be defined as, for example, the weights' absolute values or their gradients. Specifically, magnitude weight pruning refers to removing the weights closest to zero [27]. For BERT-like models, pruning can be performed in conjunction with pre-training or fine-tuning. Pruning can also be applied post-training, using, for instance, iterative magnitude pruning [35]. It refers to iteratively removing a proportion of the smallest magnitude weights and then continue fine-tuning to recover potential loss in performance. The training overhead for this iterative process is generally small; performing it on RoBERTa, 99.5% of the original validation accuracy is generally recovered in substantially less than one epoch.

2.7 Evaluation Metrics

There are several metrics that may be appropriate for evaluating the performance of HMTC solutions. Three of the most commonly used ones are micro-averaged precision, recall and F_1 score, which have also been recommended specifically for hierarchical classification tasks [59]. These metrics are extensions of the regular precision, recall and F_1 score metrics which, in turn, express: the proportion of predicted correct labels to the total number of actual labels, the proportion of predicted correct labels to the total number of predicted labels and the harmonic mean of precision and recall [60]. Contrary to macro-averaging, where metrics are computed individually for class labels and then averaged, micro-averaging computes metrics globally over all class labels and instances. Consequently, macro-averaged metrics are more influenced by minority classes, whilst micro-averaged metrics are more affected by majority classes. Following [60], micro-averaged precision, recall and F_1 score are defined in Equation

2.9, 2.10 and 2.11. For a set of n examples and k classes, Y_i^j are the ground truth labels and Z_i^j are the predicted labels. Hereafter, any mention of precision, recall or F_1 score refers to the micro-averaged version of each metric, if not stated otherwise.

$$\text{Micro-averaged precision, } P^\mu = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Z_i^j} \tag{2.9}$$

$$\text{Micro-averaged recall, } R^\mu = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j} \tag{2.10}$$

$$\text{Micro-averaged } F_1 \text{ score, } F_1^\mu = \frac{2\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j + \sum_{j=1}^{k}\sum_{i=1}^{n} Z_i^j} \tag{2.11}$$

The above metrics are all based on a notion of partial correctness. A more strict metric is exact match ratio, also called subset accuracy, that is based on a notion of absolute correctness. Specifically, exact match ratio expresses the proportion of examples whose predicted labels are identical to the ground truth labels, thereby capturing how well labels are selected in relation to each other [52]. Following [60], exact match ratio is defined in Equation 2.12, where I denotes the indicator function.

$$\text{Exact match ratio, } MR = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{2.12}$$

3 Method

In order to answer the research questions specified in Section 1.3, several activities and experiments were conducted. This included creating news article datasets and implementing solutions for training, evaluating and compressing classifiers, as well as subsequently performing these steps with multiple model types. The following chapter describes and motivates this process in detail.

3.1 Datasets

Two datasets were created for this thesis: one labeled set for fine-tuning and one unlabeled set for further pre-training. The datasets are disjoint and comprise of newspaper content published by various Bonnier News brands during the period February 2020 to February 2021. Both datasets include a variety of text retrieved from Bonnier News' data warehouse, such as news articles, editorials, feature stories, reviews, letters to the editor and recipes. All further usage of the term article refers to any type of news text in the datasets, unless otherwise stated. As the articles are relatively new and frequently published behind paywalls, the datasets are not publicly available. However, extensive dataset statistics can be found in Appendix A.

3.1.1 Preprocessing

When creating the datasets, a choice was made to exclude articles with very small or very large text bodies as they were considered outliers and as such, not representative examples of the data. This was done by removing articles shorter than 50 words – which corresponds to approximately three sentences – and articles longer than 1,500 words from the datasets. The excluded articles were commonly poorly formatted and of a certain type, such as descriptions of videos, text accompanying image compilations and long lists of real estate sales or sports results. Moreover, it is common for brands within Bonnier News to cross-publish identical or near-identical articles. In order to deduplicate the datasets, the MinHash algorithm [51] was used with the first 1,000 characters of each article to enable efficient pairwise similarity comparisons. If a pair of articles generated a Jaccard index greater than 0.9, one of the articles was discarded. Thereafter, the remaining articles in the fine-tuning and pre-training datasets underwent the same preprocessing procedure. Some articles were originally formatted in

HTML and were accordingly parsed to running text. A few articles also included auxiliary information at the end of the text, such as email and web addresses. Since such addresses are not intrinsic to an article and would not generate meaningful tokens, the sentences containing them were removed together with all subsequent text to avoid abruptly truncated sentences and gaps in the text.

The learned WordPiece vocabulary for the Swedish pre-trained BERT and ELECTRA was used to identify characters in the datasets that would generate [UNK] tokens, that is, unknown characters that are not included in the vocabulary. By manually studying these characters, it was discovered that some articles contained one or multiple sentences in languages written with non-Latin alphabets, such as Arabic and Russian. Such sentences would be tokenized to long sequences of [UNK] tokens; therefore, articles containing a large amount of non-Latin script characters were removed from the dataset. There was also a substantial amount of emojis present in the datasets that were not part of the vocabulary. Due to the difficulty of making universal replacements for emojis, all articles including emojis were excluded. Additionally, it was found that there were multiple Unicode versions of the same characters in the datasets, most notably quotation marks, hyphens and bullet points. In an effort to normalize the text and, again, avoid an abundance of [UNK] tokens, character duplicates were replaced with a single corresponding version from the vocabulary.

Lastly, the learned WordPiece vocabulary was used to tokenize and subsequently vectorize the articles. The MSL for BERT and ELECTRA – 512 tokens including the [CLS] and [SEP] tokens – was consistently used. However, in the fine-tuning and pre-training dataset, 36% and 26% of the articles, respectively, originally exceeded this length and were truncated by retaining only the first 510 tokens of the text. Even though other truncation methods can be more favorable [64], keeping only the head of the text was motivated by longer articles often having a lead paragraph that effectively functions as a synopsis of the article content. Articles shorter than 510 tokens were padded with the special token [PAD] up to the MSL. After all preprocessing steps, the size of the fine-tuning dataset was reduced by 10%, while the size of the pre-training dataset was reduced by 14%.

3.1.2 Category Taxonomy

As mentioned in Section 1.1, there is an interest at Bonnier News in examining the viability of automatic news article categorization according to a newly developed category taxonomy. However, as this taxonomy was recently completed, it has presently not been used to label enough articles to build a sufficiently comprehensive dataset. Instead, this thesis utilizes articles labeled according to an earlier taxonomy, the Category Tree for Swedish Local News, that has been developed and used by local newsrooms within Bonnier News.[1] This category tree, similarly to the newly developed taxonomy, is based on the IPTC Media Topics, which is a comprehensive standard taxonomy for categorizing news text.[2] Consequently, the taxonomies share many characteristics and this thesis employs the Category Tree for Swedish Local News as a proxy for Bonnier News' new category taxonomy.

The Category Tree for Swedish Local News, as the name suggests, structures news categories in a tree. It holds approximately 1,600 categories distributed across five hierarchical levels, where higher-level categories are more general and lower-level categories are more specific. Categories are represented by codes, which are composed of groups of three uppercase ASCII letters separated by hyphens. All category codes start with the group *RYF*, representing the root node of the tree. The number of groups in a code conveys at what level in the tree the

[1] https://github.com/mittmedia/swedish-local-news-categories
[2] https://iptc.org/standards/media-topics/

corresponding category exists and the prefix to the last group indicates its parent category. For example, the category for literature has the code *RYF-XKI-YFJ*, which informs that it is a second-level category and that its parent is the top-level category with the code *RYF-XKI*, namely culture and entertainment. Analogously, all children to the literature category will be third-level categories with codes in the format *RYF-XKI-FEY-****, such as *RYF-XKI-YFJ-HKG* for poetry. All categories have exactly one parent and inner-node categories may have multiple children. Furthermore, leaf-node categories occur at variable depths in the tree. There are a few distinct aspects of how the taxonomy is utilized:

- All articles are labeled with at least one category.
- If an article is labeled with a given category, it is also labeled with all of the ancestors to that category.
- An article may be labeled with multiple categories at each hierarchy level.
- The most specific category label for an article may be at any level in the hierarchy.

Of the roughly 1,600 categories in the taxonomy, some have never, or very rarely, been used by newsrooms within Bonnier News. Therefore, when constructing the labeled fine-tuning dataset, a decision was made to only include categories that have been used to label at least 100 articles. The motivation was to have a reasonable representation of all included categories, thus avoiding few- and zero-shot learning scenarios. Consequently, only a subset of the Category Tree for Swedish Local News was used as article labels. This subset includes 545 categories distributed across four hierarchical levels: 17, 102, 247 and 179 categories at level one, two, three and four, respectively. A comprehensive list of the categories in the subset can be found in Appendix C.

3.1.3 Fine-Tuning Data

The fine-tuning dataset comprises of 127,161 articles that have been manually labeled by journalists according to the aforementioned subset of the Category Tree for Swedish Local News. The articles come from 35 different local news brands and the dataset is highly imbalanced, with the most frequent category occurring 30,531 times and the least frequent category occurring 102 times. Furthermore, it varies greatly how many categories the articles are labeled with. The maximum number of categories used to label an article is 46 and the minimum number is one, with 5.1 categories used on average. In the dataset, the labels are represented as one-hot-encoded vectors. Additionally, the dataset has a number of metadata fields, a majority of which are derived from [44]. The metadata associated with each article is the: newspaper brand, number of images, number of authors, number of words in title, number of words in text body, mean word length in text body, median word length in text body and length of longest word in text body. The newspaper brands are one-hot-encoded and the other features are scalars. Collectively, the metadata for an article is represented as a single 42-dimensional vector. The dataset is split into training, validation and test in the ratio of 64%, 16% and 20%, resulting in 81,384, 20,345 and 25,432 articles in each set, respectively. There are five versions of the dataset, differing merely in how many categories they include. One version includes all categories represented as 545-dimensional vectors and the four other versions each include all categories from a single hierarchy level, represented as 17-, 102-, 247- and 179-dimensional vectors, respectively.

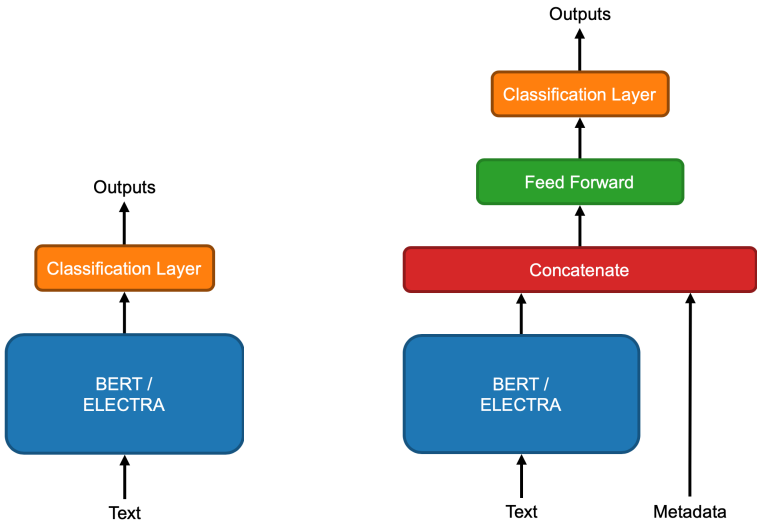
3.1.4 Pre-Training Data

The pre-training dataset comprises of 240,672 unlabeled articles from 46 newspaper brands, including national newspapers and trade magazines. There were two reasons for including articles from more brands than those in the fine-tuning dataset. Firstly, it allowed for more

pre-training data to be used. Secondly, the resulting models are more practically applicable compared to if they would be specialized solely on a local news domain. For further pre-training of BERT, the article texts needed to be segmented into sentences. Thus, a second version of the dataset was created by performing sentence segmentation on the articles using a Swedish pre-trained UDPipe model [61].

3.2 Models

Two main model architectures based on pre-trained contextualized language models were created to be trained for HMTC, both presented in Figure 3.1. These architectures were built on top of either the Swedish pre-trained BERT or ELECTRA discriminator released by KBLab. Specifically, the cased, base-size versions of the models were used, each one having 12 layers, a hidden size of 768 and 12 self-attention heads. They were downloaded from KBLab's model repository on the Hugging Face website[3] and will henceforth be referred to as KB-BERT and KB-ELECTRA, respectively. The model architectures described below were created identically for all of the pre-trained models.



(a) Basic architecture (b) Metadata architecture

Figure 3.1: Model architectures

3.2.1 Basic Architecture

The most basic architecture was created by adding a classification head on top of the pre-trained model, similar to how previous works have adapted BERT for TC [20, 3, 64]. This was done by introducing a linear layer, with an output size equal to the number of classes, over the final hidden state corresponding to the [CLS] token. Thereafter, a sigmoid activation function was applied to the output logits to produce independent probabilities for each class label. In practice, when BERT is trained on NSP, the aggregate sequence representation is learned by adding a linear layer with a tanh activation after the final [CLS] state, sometimes colloquially referred to as a pooling layer.[4] It is the output from this pooling layer that is normally used as input to a classification layer. Since ELECTRA does not employ any sequence-level pre-training task, these additional weights do not exist in the pre-trained model. Consequently,

[3]https://huggingface.co/KB
[4]https://github.com/google-research/bert/blob/master/modeling.py

a pooling layer identical to the one in KB-BERT, but with randomly initialized weights, was added to KB-ELECTRA prior to fine-tuning. Some implementations use slightly differing pooling strategies when adapting ELECTRA for TC, for instance, by adding multiple linear layers with GELU activations.[5] However, there is seemingly no consensus on which method is favored, so for sake of direct comparability, the exact same architecture was used for KB-BERT and KB-ELECTRA.

3.2.2 Metadata Architecture

Inspired by previous work [44], the basic architecture was extended to support the inclusion of metadata features. This was done by concatenating the final hidden state corresponding to the [CLS] token from the language model with a metadata feature vector, resulting in a new document representation. Following this concatenation, two fully connected layers were added, each with a hidden size of 1048 and a ReLU activation function. Lastly, the same classification layer as in the basic architecture was added to output the final class label probabilities.

3.2.3 Global and Local Classifiers

Global classifiers were created with the basic architecture by training a single classifier over all categories. Consequently, the output size of all global classifiers was 545. Both model architectures were also used to create local classifiers. Specifically, an LCL approach was used, as it was the only feasible local classifier approach given the amount of available computational resources. Four local classifiers, one for each hierarchy level, were created with the architectures, using the number of classes at the corresponding level as output size. When training local classifiers, a HTrans strategy was used to recursively transfer all model parameters, except the final layer weights, from parent to child classifiers.

3.3 Training

The process of training the classifiers involved fine-tuning on labeled data and, for some models, further pre-training on unlabeled data.

3.3.1 Further Pre-Training

KB-BERT and KB-ELECTRA were domain-specialized by performing additional pre-training on the pre-training dataset. Scripts from the official BERT repository on GitHub[6] were used for generating the final training examples and performing pre-training of KB-BERT. As the scripts are implemented in TensorFlow, the TensorFlow checkpoint for KB-BERT was used for training and later converted to a PyTorch model to be compatible with the fine-tuning implementations. For generating the final training examples and pre-training KB-ELECTRA, a PyTorch re-implementation[7] of the ELECTRA pre-training was utilized. This reimplementation has previously been used to replicate the results of the original ELECTRA paper, and it is the most starred repository of its kind on GitHub. For this thesis, the pre-training script was slightly modified, so that both the discriminator model, here referred to as KB-ELECTRA, and its associated generator model from KBLab were used to initialize model parameters.

Following the guidelines in the official BERT repository, KB-BERT and KB-ELECTRA were further pre-trained for 100,000 steps using a learning rate of 2e-5. It is likely that even more pre-training could be beneficial for downstream performance [33, 12], but due to resource

[5] https://huggingface.co/transformers/_modules/transformers/models/electra/modeling
[6] https://github.com/google-research/bert
[7] https://github.com/richarddwang/electra_pytorch

constraints, this was not investigated. Pre-training with longer sequences is disproportionally expensive due to the self-attention mechanism, which is why the original BERT models were pre-trained with an MSL of 128 for 90% of the steps, using an MSL of 512 only for the final 10% of the steps [20]. Similarly, here, KB-BERT and KB-ELECTRA were initially further pre-trained for 90,000 steps using an MSL of 128 and a batch size of 32, whereafter training resumed for an additional 10,000 steps with an MSL of 512 and a batch size of 8. The selection of these hyperparameters was constrained by the memory capacity of the graphics processing unit (GPU) model utilized for training. Henceforward, the domain-specialized versions of KB-BERT and KB-ELECTRA will be referred to as KB-BERT^{S} and KB-ELECTRA^{S}.

3.3.2 Fine-Tuning

The models described in Section 3.2 were trained and evaluated on the fine-tuning dataset by minimizing the binary cross-entropy loss. Initially, global classifiers using the basic architecture with KB-BERT and KB-ELECTRA were trained. These fine-tuned global models will be referred to as BERT$_{G}$ and ELECTRA$_{G}$. Thereafter, the basic architecture with KB-BERT and KB-ELECTRA were trained with the LCL approach. These two groups of four fine-tuned local classifiers will be referred to as BERT$_{L}$ and ELECTRA$_{L}$. Using the basic architecture, KB-BERT^{S} and KB-ELECTRA^{S} were used to train global classifiers, which will referred to as BERT$^{S}_{G}$ and ELECTRA$^{S}_{G}$. Lastly, the pre-trained model utilized as basis for the top-performing global classifier in terms of F_1 score was used with the metadata architecture and trained with the LCL approach. This classifier will be referred to as META$_{L}$.

Hyperparamers and Optimization

All classifiers were trained using largely the same hyperparameters, with fine-tuning settings inspired by the original BERT and ELECTRA papers [20, 15]. An MSL of 512 and a batch size of 32 was used. The classifiers were trained for a maximum of four epochs utilizing dropout regularization with probability 0.1. For optimization, the AdamW algorithm [37] was used with 0.01 weight decay and linear learning rate decay after warmup on 10% of the training data. Additionally, similar to [44], an SCut strategy was used to tune the classification thresholds on the validation set. Since SCut is problem-specific, all classifiers were also evaluated on the validation set using 0.5 thresholds for all classes. If this resulted in better performance than when using the tuned thresholds, all thresholds were set to 0.5 instead. The learning rate was tuned with BERT$_{G}$ on the validation set and selected from {1e-4, 5e-5, 2e-5}. To ensure that it was not overly biased towards BERT$_{G}$, ELECTRA$_{G}$ and the top-level local classifiers for BERT$_{L}$, ELECTRA$_{L}$ and META$_{L}$ were all trained for – and subsequently evaluated after – one epoch, using each one of the three candidate learning rates. If a model yielded better results after one epoch with a different learning rate than the one selected for BERT$_{G}$, this was used for the remainder of the training. For BERT$^{S}_{G}$ and ELECTRA$^{S}_{G}$, the learning rates were the same as for BERT$_{G}$ and ELECTRA$_{G}$, respectively.

3.4 Compression

Post the fine-tuning, BERT$_{G}$'s and ELECTRA$_{G}$'s robustnesses to compression were evaluated, largely inspired by [35]. The models were quantized from FP32 to FP16 and to INT8, respectively, employing a dynamic quantization method. This involved quantizing all model parameters in advance except for the activations, which were kept at full precision and dynamically quantized during inference. This method is favorable for Transformer-based models, where execution time is dominated by loading parameters from memory.[8] Subsequently, iterative magnitude pruning [45] was applied to the models, considering sparsity levels of

[8] https://pytorch.org/docs/stable/quantization.html

15%, 30%, 45% and 60%. Iteratively, 15% of the smallest weights were pruned, whereafter fine-tuning resumed until 99.5% of the F_1 score on the validation data was recovered or a full training epoch was completed. The pruned models were evaluated after training on every 10% portion of the data. For this recovery training, it was necessary to reduce the batch size to 16 because the models required associated tensors that held pruning masks, which temporarily increased the GPU memory usage. Pruning was performed by zeroing out weights, which allowed for evaluating the theoretical effect of pruning but did not reduce memory footprints or inference times for the models in practice.

3.5 Baseline

For additional comparison, an LCN approach with SVMs was used, which is a relatively simple yet adequate baseline for HMTC problems [4, 52]. Specifically, a one-versus-rest strategy was employed to train one linear SVM classifier per category, resulting in 545 SVMs in total. The SVMs were trained on the TF-IDF representations of the full-length article texts in the fine-tuning dataset. The regularization parameter C was tuned on the validation set from the interval [1e-2, 1e2] using randomized search [10].

3.6 Evaluation

Classification performance was evaluated in terms of exact match ratio and micro-averaged precision, recall and F_1 score, as defined in Section 2.7. For all models, evaluation was performed on the fine-tuning test set containing all class labels. The F_1 score was the most prioritized metric and was used for hyperparameter tuning and threshold optimization. To avoid class-membership inconsistencies, the outputs from all classifiers were post-processed with a class-prediction top-down approach, discarding any predicted class labels that had one or more ancestor labels missing from the output. Due to resource constraints, it was infeasible to evaluate every classifier across multiple training runs to account for variability of the results. Nevertheless, to investigate some level of classification robustness, BERT_G was trained and evaluated across three random seeds. From these results, the mean and sample standard deviation were computed, where the latter was assumed to approximately hold for all other Transformer-based classifiers.

3.7 Experimental Setup

The code base for this thesis was implemented in Python 3.8.3. PyTorch [46] and Hugging Face Transformers [72] were used for creating and using all the models, with the exception of the original pre-training script for BERT, which uses TensorFlow [1]. PyTorch APIs were also used for performing quantization and pruning. Additionally, Scikit-learn [47] implementations of TF-IDF, SVMs, randomized search and metrics were used. Sentence segmentation of the BERT pre-training data was performed with spaCy-UDPipe, Beautiful Soup was utilized for HTML parsing and the SnaPy implementation of MinHash was used. A complete list of the used libraries and their versions can be found in Appendix B. Training was run on virtual machines (VMs) on Google Compute Engine, employing predefined VM images for DL with NVIDIA Tesla T4 GPUs with 15 GB memory each. Four GPUs were utilized for fine-tuning, where training was parallelized with PyTorch Distributed Data Parallel. The pre-training scripts only supported the usage of a single GPU. All experiments were run with a fixed seed, except for the two additional training runs of BERT_G.

4 Results

This chapter summarizes and describes the results of the experiments detailed throughout Section 3.3 to 3.6.

4.1 Hyperparameter Tuning

Learn. Rate	Epoch	Train. Loss	Val. Loss	MR	P^{μ}	R^{μ}	F_1^{μ}
1e-4	1	0.0797	0.0306	0.0141	0.8134	0.3436	0.4831
	2	0.0256	0.0220	0.1197	0.8188	0.4514	0.5820
	3	0.0197	0.0191	0.1823	0.8149	0.5087	0.6263
	4	**0.0167**	**0.0183**	**0.2062**	0.8077	**0.5363**	**0.6446**
5e-5	1	0.1306	0.0429	0.0000	0.0000	0.0000	0.0000
	2	0.0417	0.0377	0.0018	0.8327	0.2206	0.3488
	3	0.0343	0.0315	0.0134	0.8142	0.3299	0.4696
	4	0.0305	0.0298	0.0224	0.8144	0.3591	0.4984
2e-5	1	0.0994	0.0388	0.0011	**0.8329**	0.1918	0.3118
	2	0.0320	0.0269	0.0615	0.8251	0.4001	0.5389
	3	0.0243	0.0227	0.1144	0.8113	0.4585	0.5858
	4	0.0211	0.0215	0.1382	0.8144	0.4752	0.6002

Table 4.1: Learning rate tuning on validation set with BERT_{G}

In Table 4.1, the results of the learning rate tuning with BERT_{G} are presented, with the best values marked in bold. It should be noted that the difference between certain metric values, such as the two highest precisions, is negligible and could be a result of randomness in the training process. The best results for every loss and metric, except precision, are obtained after training for four epochs with a learning rate of 1e-4. For all learning rates, the training and validation losses decrease – while the F_1 score increases – monotonically with the number of epochs, yielding the best results after four epochs. Accordingly, all other classifiers were trained for four epochs. Additionally, as seen in Figure 4.1, the metrics change similarly over time for all learning rates, so that the one producing the highest F_1 score after the first

epoch also produces the highest F_1 score after the last epoch. Consequently, for the remaining models that are tested, the learning rate producing the highest F_1 score after a single training epoch is selected. For ELECTRA_G, 1e-4 also gives the best results. Accordingly, 1e-4 is used for $\text{BERT}^\text{S}_\text{G}$ and $\text{ELECTRA}^\text{S}_\text{G}$ as well. For all top-level local classifiers, that is for BERT_L, ELECTRA_L and META_L, a learning rate of 5e-5 yields the best results after one epoch.

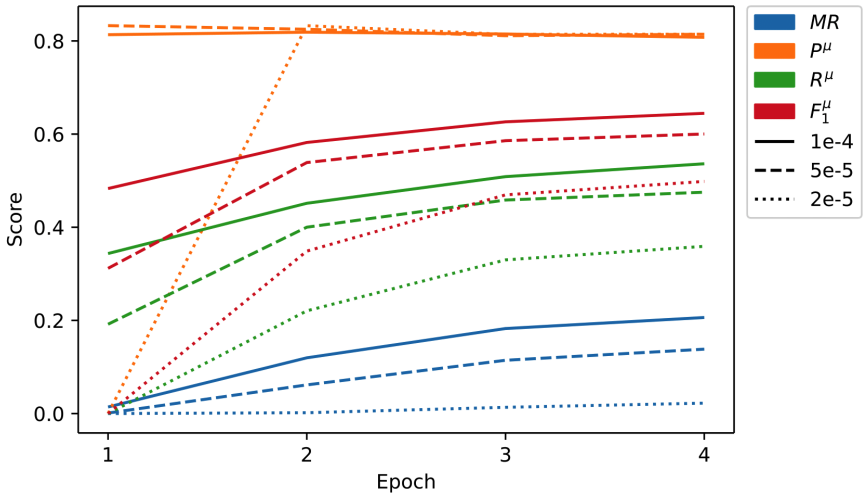


Figure 4.1: Change of evaluation scores on validation set over number of epochs

It may be observed that the training loss is occasionally larger than the corresponding validation loss. This is because the training loss is cumulatively computed during each epoch, while the validation is calculated after each epoch. Additionally, dropout regularization is applied during training, which influences the training loss negatively. As such, the training and validation losses are only comparable relative to themselves, not to each other. The precision is relatively stable around 0.8 for all learning rates during training, while the recall is considerably lower but steadily improves during training. Notably, using learning rate 5e-5 and training for one epoch, all metrics are zero. The reason is that the classifier does not assign any class labels to any of the articles in the validation set.

For all global classifiers, the best F_1 scores on the validation set are obtained using the thresholds tuned with SCut. Using the LCL approach however, the results are more ambiguous. For BERT_L and META_L, the local classifiers at the first and second levels, as well as the top-level classifier in ELECTRA_L, produce higher F_1 scores using 0.5 thresholds. Therefore, during testing, 0.5 is used as threshold for the categories predicted by these local models. Generally, when the tuned thresholds perform better on the validation set, it is because they generate higher recall compared to when employing 0.5 thresholds.

4.2 Classification Performance

The evaluation scores for all classifiers on the test set are shown in Table 4.2. For BERT_G, the mean and sample standard deviation of each metric is reported. The F_1 score has a relative standard deviation of 0.3%, which is assumed to hold approximately for the other Transformer-based classifiers. Comparing the first global classifiers to the SVM baseline, BERT_G performs slightly better, while ELECTRA_G performs significantly worse. The LCL approaches BERT_L and ELECTRA_L both outperform their global counterparts, but with ELECTRA_L yielding worse results than the baseline. The pre-training implementation utilized for KB-BERT^S allows for evaluation on an automatically generated validation set. The resulting MLM loss is 4.5572 and the NSP loss is 0.2305, with corresponding accuracies of

0.3275 and 0.9050. The pre-training implementation used for KB-ELECTRA^{S} outputs only the final training loss, which is 9.3219. After fine-tuning, ELECTRA$^{S}_{G}$ performs better than ELECTRA$_{G}$. BERT$^{S}_{G}$ performs negligibly worse than BERT$_{G}$, with a performance difference that can be attributed to randomness. Of the global classifiers, the one built on KB-BERT produces the highest F_1 score and is accordingly used as basis for training the local classifiers using metadata features, META$_{L}$. META$_{L}$ performs best across all metrics, except for precision.

Model	MR	P^{μ}	R^{μ}	F_1^{μ}
SVM	0.2752	**0.8250**	0.5076	0.6285
BERT$_{G}$	0.1919±0.0047	0.7998±0.0037	0.5251±0.0032	0.6340±0.0019
ELECTRA$_{G}$	0.0706	0.8023	0.3940	0.5285
BERT$_{L}$	0.2638	0.7799	0.5765	0.6629
ELECTRA$_{L}$	0.2031	0.7863	0.5121	0.6203
BERT$^{S}_{G}$	0.1860	0.7994	0.5241	0.6331
ELECTRA$^{S}_{G}$	0.0913	0.8025	0.4261	0.5566
META$_{L}$	**0.2954**	0.7852	**0.5836**	**0.6695**

Table 4.2: Evaluation scores on test set

In Table 4.3, the classifiers' performance per hierarchy level in the category taxonomy are shown. The metrics are micro-averaged over the categories at each level and the best values for each classifier are bolded. All classifiers have the highest recall and F_1 score for the categories at the top-most level. Table 4.4 displays statistics on the output predictions of each classifier on the test set. The second column refers to the number of categories that a classifier never assigned to a test example, and the third column refers to the average number of predicted categories per article. For reference, the true statistics of the test set are included in the top row, and the closest values in each column are bolded. All classifiers disregard some categories and, on average, predict fewer labels compared to the ground truth. ELECTRA$_{G}$ and ELECTRA$^{S}_{G}$, in turn, exclude the most categories and predict the smallest number of categories on average.

4.3 Effects of Compression

Table 4.5 displays the evaluation scores on the test set after compressing BERT$_{G}$ and ELECTRA$_{G}$. The second column indicates either the quantization data type or the pruning sparsity level. Additionally, the second-to-last column shows the relative change in F_1 score compared to the uncompressed models. For the pruned models, the last column displays the amount of recovery training that was performed at each pruning iteration, expressed in fractions of a full training epoch. After quantization to FP16, both classifiers virtually produce the same results as their full-sized counterparts. When the target data type is INT8, performance degrades for both models, more so for BERT$_{G}$ than for ELECTRA$_{G}$. After pruning 15%, 30% and 45% of the smallest weights, both models produce F_1 scores that are within 1% of the original metric value on the test set. After 60% of the weights have been pruned, performance markedly degrades for both models, particularly for ELECTRA$_{G}$. This coincides with the recovery training reaching the maximum of a full epoch without having recovered 99.5% of the original F_1 score on the validation set. For the other sparsity levels, the performance on the validation set is recovered in significantly less than one epoch.

Model	Level	P^{μ}	R^{μ}	F_1^{μ}
SVM	1	0.8182	**0.6479**	**0.7231**
	2	0.8162	0.4988	0.6192
	3	0.8382	0.4279	0.5666
	4	**0.8406**	0.4455	0.5823
BERT_{G}	1	**0.8180**	**0.6957**	**0.7519**
	2	0.7967	0.5282	0.6353
	3	0.7905	0.4368	0.5627
	4	0.6973	0.3045	0.4239
$\text{ELECTRA}_{\text{G}}$	1	**0.8229**	**0.6007**	**0.6945**
	2	0.7919	0.4192	0.5482
	3	0.8111	0.2567	0.3900
	4	0.6263	0.1064	0.1819
BERT_{L}	1	0.7846	**0.7363**	**0.7597**
	2	0.7664	0.5808	0.6608
	3	**0.7998**	0.4557	0.5806
	4	0.7638	0.4275	0.5482
$\text{ELECTRA}_{\text{L}}$	1	0.7755	**0.7176**	**0.7454**
	2	0.7941	0.4978	0.6120
	3	**0.8182**	0.3637	0.5036
	4	0.7471	0.3579	0.4840
$\text{BERT}_{\text{G}}^{\text{S}}$	1	**0.8236**	**0.6896**	**0.7506**
	2	0.7998	0.5264	0.6349
	3	0.7860	0.4421	0.5659
	4	0.7100	0.2804	0.4021
$\text{ELECTRA}_{\text{G}}^{\text{S}}$	1	**0.8203**	**0.6411**	**0.7197**
	2	0.7953	0.4413	0.5676
	3	0.8071	0.2970	0.4343
	4	0.6548	0.1257	0.2110
META_{L}	1	0.7851	**0.7385**	**0.7611**
	2	0.7649	0.5790	0.6591
	3	0.8071	0.4617	0.5874
	4	**0.8086**	0.4735	0.5973

Table 4.3: Evaluation scores per hierarchy level on test set

Model	# Non-Used Labels	Avg. # Labels
Test Set	0	5.10
SVM	**5**	3.21
BERT_{G}	67	3.38
$\text{ELECTRA}_{\text{G}}$	293	2.50
BERT_{L}	44	3.77
$\text{ELECTRA}_{\text{L}}$	110	3.32
$\text{BERT}_{\text{G}}^{\text{S}}$	68	3.34
$\text{ELECTRA}_{\text{G}}^{\text{S}}$	202	2.71
META_{L}	21	**3.79**

Table 4.4: Statistics on output predictions

Model	Method	Type	MR	P^{μ}	R^{μ}	F_1^{μ}	ΔF_1^{μ} (%)	Epoch
BERT_{G}	Quantization	FP16	0.1956	0.7958	0.5280	0.6348	+0.0015	-
		INT8	0.1012	0.8437	0.4173	0.5584	−12.0341	-
	Pruning	15%	0.2001	0.7947	0.5245	0.6319	−0.4563	0.2
		30%	0.2056	0.7958	0.5300	0.6362	+0.2234	0.3
		45%	0.1953	0.7884	0.5271	0.6318	−0.4739	0.1
		60%	0.1621	0.7896	0.5007	0.6128	−3.4687	1.0
$\text{ELECTRA}_{\text{G}}$	Quantization	FP16	0.0706	0.8023	0.3940	0.5285	+0.0025	-
		INT8	0.0666	0.7634	0.3979	0.5231	−1.0095	-
	Pruning	15%	0.0675	0.8005	0.3927	0.5269	−0.2852	0.0
		30%	0.0781	0.8004	0.3992	0.5327	+0.8049	0.4
		45%	0.0717	0.8012	0.3936	0.5279	−0.1114	0.2
		60%	0.0217	0.8148	0.3276	0.4673	−11.5790	1.0

Table 4.5: Evaluation scores on test set after compression

5 Discussion

The following chapter discusses the results presented in Chapter 4, as well as the method for obtaining them, as described in Chapter 3.

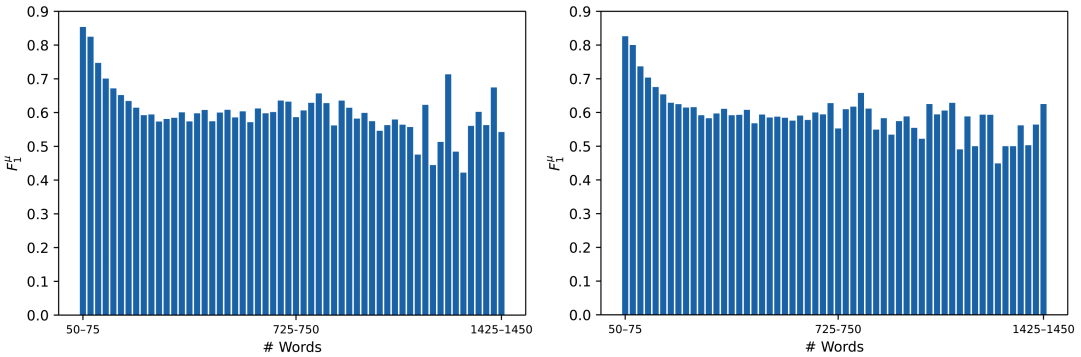
5.1 Results

A notable and somewhat surprising characteristic of the results is that none of the evaluated classifiers yield an F_1 score greater than 0.7. Although not directly comparable, this is low in relation to a lot of previous work applying BERT to TC, where it is not uncommon for F_1 scores and accuracies to have magnitudes around 0.8 or 0.9 [3, 64, 44, 26]. One reason for the comparatively weak performances could be that the studied TC problem is fairly difficult, something that is discussed in Section 5.1.7. If so, what is even more surprising, is that the SVM baseline performs on par with, or even outperforms, several of the classifiers built on BERT or ELECTRA. However, this is not unheard of for HMTC problems with several hundred classes, where local SVM-based classifiers can outperform global BERT-based classifiers, all producing F_1 scores below 0.7 [52]. Nevertheless, the findings are interesting considering that SVMs are substantially less complex and resource intensive to train than Transformer-based models.

5.1.1 Strong Baseline

Studying Table 4.2, it may be observed that the SVM baseline has the highest precision and second highest exact match ratio out of all classifiers. This indicates that the labels assigned by the baseline are relevant, and relatively often perfectly matches the ground truth labels. Additionally, as seen in Table 4.4, the baseline disregards the least amount of categories. An explanation for these observations could be that the baseline is an LCN approach, where the multi-label classification problem is decomposed into 545 individual binary classification problems. As a result, and contrary to all other models, the local SVMs do not learn anything about how many labels to assign to an article, only if it should be labeled with a specific category or not. Given the large dimension of the label space and the large variation in number of ground truth labels per article, this local focus on individual categories is likely beneficial for the baseline's performance. Another reason for the baseline to perform well could be that the complexity of the TF-IDF features is fully adequate for the task. Plausibly, many cate-

gories are associated with very specific and unique words, such as terms in sports, economy or music, which are highly discriminative for classification. TF-IDF representations are good at capturing such characteristics and it is possible that the context in which distinctive terms occur – which is what BERT and ELECTRA build representations on – is of lesser importance.



(a) SVM (b) BERT_G

Figure 5.1: F_1 score relative to article length

An explicit advantage of the baseline is that SVMs do not have an MSL, meaning that they operate on the entirety of all articles. Compared to the BERT- and ELECTRA-based classifiers, this provides the SVMs with more information about longer articles that otherwise require truncation to account for an MSL. Since 36% of the articles in the fine-tuning dataset exceed the MSL of 512 tokens, this could potentially benefit the baseline's overall performance relative to the other classifiers. In Figure 5.1, the F_1 scores for SVM and BERT_G are shown in relation to article lengths. The F_1 scores have been calculated and micro-averaged across groups of articles binned by their length, each bin ranging across 25 words. Although the baseline has comparatively high F_1 scores for some of the rightmost bins, it does not appear to be systematically stronger than BERT_G at classifying longer articles. If anything, both classifiers seem to be slightly better at handling shorter articles in general.

5.1.2 BERT Versus ELECTRA

Another somewhat surprising finding is that all ELECTRA-based classifiers are clearly outperformed by their BERT-based counterparts – especially when employing a global classifier approach. The difference in F_1 scores between the model types is largely due to the ELECTRA-based classifiers having lower recall. This can be related to the ELECTRA-based classifiers consistently neglecting more categories and assigning fewer labels per example compared to the BERT-based dittos, as seen in Table 4.4. To the author's knowledge, there is no previous research that explicitly compares BERT and ELECTRA for any type of multi-label TC. However, ELECTRA has outperformed BERT on NLP benchmarks that include binary and multi-class classification tasks [15]. The poor performance of ELECTRA-based classifiers is also confounding considering that KB-ELECTRA has been pre-trained on significantly more data than KB-BERT, a majority of which is newspaper text, as shown in Table 2.1. As there is no published details on the pre-training process or downstream performance of KB-ELECTRA, there exists no additional reference point for further comparison with KB-BERT. For the same reason, it is difficult to assess if there is some aspect of KBLab's method in particular, or the ELECTRA pre-training task in general, that might cause the ELECTRA-based classifiers to perform poorly.

A possible explanation for the performance differences between the BERT- and ELECTRA-based classifiers could be how their pooling layers are initialized. While the pooling layer of the former is initialized with pre-trained weights, the latter's pooling layer is randomly

initialized. However, this seems unlikely to cause drastically varying results. For example, when using HTrans to initialize the parameters of local classifiers, the possible disadvantage of KB-ELECTRA not having pre-trained pooling weights should reasonably decline over time, the reason being that the initialized pooling layers of a child classifier will effectively have been tuned during the training of its parent classifier. Studying Table 4.3 however, ELECTRA_L actually performs most on par with BERT_L at the very first level, indicating that the pooling weight initialization is not crucial for the classifiers' performances. It should also be acknowledged that a different pooling layer implementation, for example with GELU activations, possibly could have increased the performance of the ELECTRA-based classifiers.

5.1.3 Globality and Locality

The LCL approaches with BERT and ELECTRA both outperform their global classifier equivalents. Specifically, in terms of F_1 score, BERT_L outperforms BERT_G by 4.4% and ELECTRA_L outperforms ELECTRA_G by 17.4%. It should be noted that the performance increase for ELECTRA_L is relative to a substantially lower score than for BERT_L. For both models, the higher F_1 score is a result of increased recall at the cost of a slightly decreased precision. Additionally, the LCL approaches neglect less labels entirely and assign more labels to articles than the corresponding global approaches, as seen in Table 4.4. This can be related to the notion that the locality of the baseline classifier is likely beneficial for its performance. In a similar way, the LCL approaches seemingly benefit from the, albeit less extreme, problem decomposition that enables four individual classifiers to each focus on a subset of categories. Analogously, one might expect that an LCPN or even LCN approach would further improve the Transformer-based classifiers. However, all performance gains due to increased classifier locality should be contrasted with the increase in resource consumption. In this thesis, the improvements from using an LCL approach over a global classifier approach come at the cost of nearly quadruplicating the resources needed for training, inference and storage.

Global Model	LCL Model	Level	ΔF_1^μ (%)
BERT_G	BERT_L	1	+1.0374
		2	+4.0139
		3	+3.1811
		4	+29.323
ELECTRA_G	ELECTRA_L	1	+7.3290
		2	+11.6381
		3	+29.1282
		4	+166.0803

Table 5.1: Improvements for the LCL models relative to the corresponding global models

Studying Table 4.3 for BERT_G, ELECTRA_G, BERT_L and ELECTRA_L, the average F_1 scores steadily decrease for categories at increasingly low levels. This is reasonable, given that categories generally get less well-represented for each hierarchy level, as seen in Table A.4. Furthermore, the LCL-based classifiers outperform their global counterparts on every hierarchy level. Again, this is sensible given that the LCL approach specializes a separate model per level, as opposed to employing a single model across all levels. In Table 5.1, the relative improvement of classification performance per level using the LCL approaches compared to their global classifier equivalents is presented. Generally, utilizing local classifiers is increasingly beneficial for more specific, lower-level categories. Because more general categories are more frequent and thus, will have larger impact on the training losses of global classifiers, it is reasonable that the local classifiers are increasingly better at handling sparser categories.

Additionally, HTrans has previously been shown to be particularly effective in improving classification performance for less frequent categories [6].

5.1.4 Variable Effect of Domain Specialization

The domain-specialized ELECTRA$_{G}^{S}$ outperforms ELECTRA$_{G}$ by 5.3% in terms of F_1 score. This is in line with previous work, where further in-domain pre-training improves performance on downstream classification [44]. Contrarily, BERT$_{G}^{S}$ performs marginally worse than BERT$_{G}$, albeit with such an insignificant difference that it can be attributed to randomness. Regardless, it is evident that the further pre-training of KB-BERT did not improve performance. An explanation for this could be the sentence segmentation applied to the pre-training data for KB-BERT. Studying samples of the segmented articles, it was observed that the employed UDPipe solution did not perform flawlessly. Compared to the custom scripts used to split sentences in the original pre-training data for KB-BERT [39], the UDPipe model is likely inferior. Consequently, it is conceivable that there were undesirable discrepancies between the original and newly created pre-training data for KB-BERT. This would necessitate the model to not only adapt to a specific domain, but also learn, for example, how the new pre-training data is formatted. In addition, this might be the reason for KB-BERT^{S}s low accuracy on the MLM task.

5.1.5 Utilizing Metadata

Out of all classifiers, META$_{L}$ performs the best across all metrics and outputs the most category labels on average. Relative to the directly comparable BERT$_{L}$, the inclusion of metadata features improves the F_1 score by roughly 1%. This increase is not sizeable, but is deemed significant given the relative standard deviation of 0.3% for F_1 scores. The performance gain from using metadata features in conjunction with BERT is in line with previous work [44]. Since no ablation studies have been performed, it is difficult to assess the discriminative qualities of the individual metadata features. Considering Figure A.13 to Figure A.20 however, it is likely that features such as article length, brand and number of images influence classification more, as they vary the most across categories.

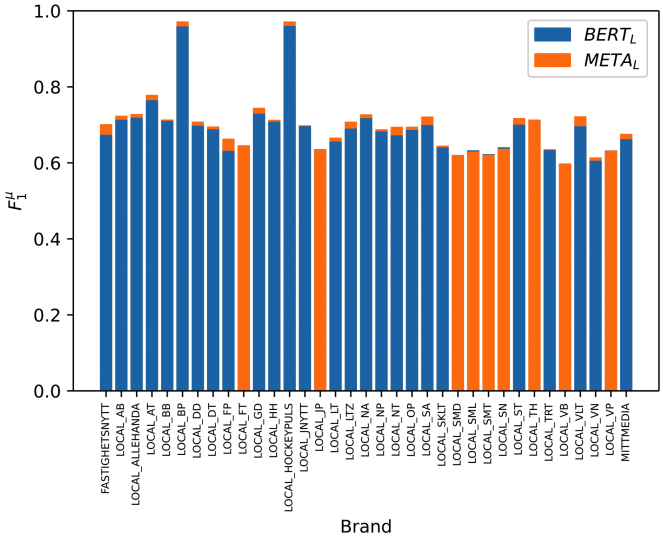


Figure 5.2: F_1 score per brand for BERT$_{L}$ and META$_{L}$

In Figure 5.2, the micro-averaged F_1 scores per brand are shown for BERT$_{L}$ and META$_{L}$. For each brand, the bar in front is always the lowest of the two, meaning that if only one bar is visible, the F_1 scores are nearly identical for both classifiers. Notably, META$_{L}$ performs better

for a majority of the categories, and where it performs worse, it is only by a small margin. This can be attributed to the classifier learning combinations of textual content and metadata that are characteristic for individual brands. It should be added that not only BERT_L and META_L, but all other classifiers as well, perform considerably better on articles from the brands *LOCAL_BP* and *LOCAL_HOCKEYPULS*. These are brands specialized on bandy and hockey news, respectively, and are likely to have articles written in similar styles with labels from a small subset of subject-specific categories. Additionally, as seen in in Figure A.12, there are relatively few articles from each of the brands in the dataset. Jointly, this could be the reason for the anomalistically high F_1 scores for these two brands.

5.1.6 Compression Robustness

Quantizing BERT_G and ELECTRA_G to FP16, both classifiers practically retain their original performance, as seen in Table 4.5. Similar findings have been reported when quantizing RoBERTa to FP16 [35]. Lowering the target bit-resolution to INT8, the performances of both models drop, particularly for BERT_G. Apart from noting that the smaller drop for ELECTRA_G is relative to a lower original score, it is hard to speculate about why BERT_G is less robust to quantization in this case. Nonetheless, this larger reduction is not extraordinary, given that it has previously been demonstrated how BERT's performance can drop by up to 10% following dynamic INT8 quantization [57].

Generally, both classifiers are robust to pruning, maintaining their performances after removing 15%, 30% and 45% of the weights. That the test scores occasionally increase as a result of pruning is not meaningful, since the 10% evaluation intervals during recovery training can cause the pruned models to overshoot 99.5% of the original validation F_1 scores. Nonetheless, it is noteworthy that the classifiers' recovery training last for substantially less than one epoch for these sparsity levels. This indicates that further recovery training is likely to improve upon the performance of the original classifier by several percent. However, this should be contrasted with any corresponding performance increase when training the full-sized model for an equal amount of time. When both classifiers reach sparsity levels of 60%, the test scores decrease visibly, most significantly for ELECTRA_G. Concurrently, the recovery training lasts for an entire epoch without reaching the threshold for the validation F_1 score. Again, if desirable, more recovery training could likely improve the results of the pruned classifiers. Overall, the effects and dynamics of pruning are similar to those reported for RoBERTa and BERT in previous work [35, 27]. The compression results are collectively interesting, indicating that up to 50% of the information stored in a Transformer-based classifier could be disposable. Additionally, this redundancy can be discarded with low computational overhead, by reducing the bit resolution of weights or removing weights altogether.

5.1.7 Error Analysis

A consistent pattern across all classifiers is that they have low recall relative to precision. Moreover, all classifiers systematically produce fewer labels on average than the ground truth. In Figure 5.3, the false negative rate (FNR) and false positive rate (FPR) for each classifier is presented. The rates are averaged across all categories and the vertical axis is logarithmically scaled. FNR is the proportion of incorrectly excluded ground truth labels from the classification output, while FPR is the proportion of erroneous labels included in the classification output. Evidently, the FNR is consistently much higher than the FPR, where the latter never exceeds 2e-3. Collectively, this means that the classifiers, rather than predicting too many labels where some are incorrect, are prone to predicting too few labels and miss some of the correct ones. An extreme effect of this can be observed for BERT_G in Table 4.1, where one classifier outputs no labels.

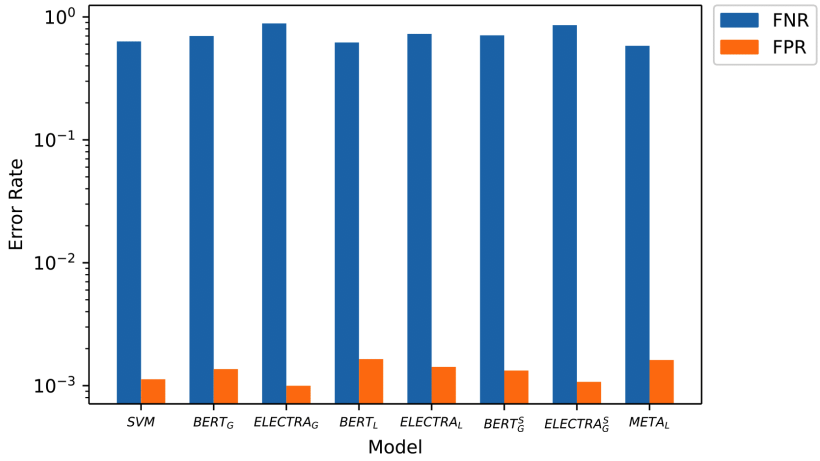


Figure 5.3: Error rates

Considering that the employed class-prediction top-down approach enforces class-membership consistency by removing labels, the post-processing could be the cause for these observations. However, averaged across all classifiers, the post-processing step merely reduces the number of output labels by 3.4%. More likely, the overall low recall and high FNR originate from characteristics of the fine-tuning dataset. As previously mentioned, an average article is sparsely labeled in relation to the total number of categories: 5.1 labels out of 545 possible ones. Additionally, even though the range of number of ground truth labels is wide, a majority of the articles have relatively few labels, as seen in Figure A.2. Presumably, this is the main explanation for the type of classification errors observed.

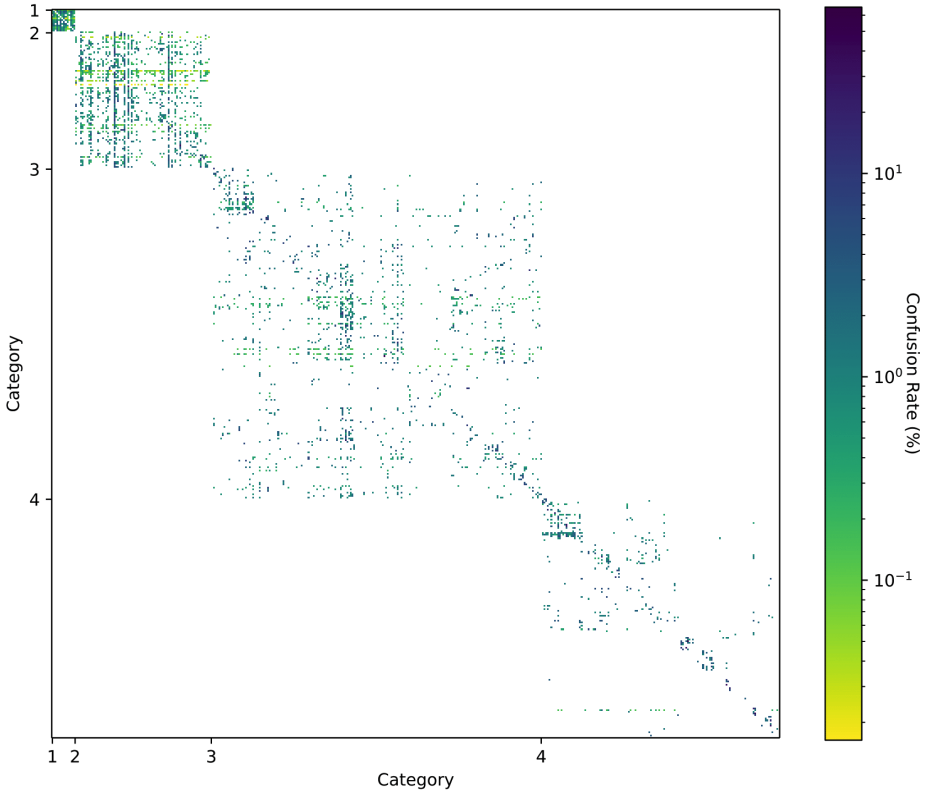


Figure 5.4: Heat map of pairwise confusion rates for $BERT_G$

To further investigate the classification errors, it is possible to study which categories the classifiers most commonly confuse with each other. To this end, a pairwise confusion rate is computed between all categories at the same hierarchy level. For each category pair at a given level, the number of test examples where the first category is falsely rejected while the second category is erroneously predicted are counted. The pairwise confusion rate is then defined as this count normalized by the frequency of the first category in the test set. To provide an example, Figure 5.4 displays the confusion rates for BERT_G in a heat map. The confusion rates are computed from the test-set predictions prior to post-processing. Along the axes, categories appear in the same order as in Appendix C, so that categories with the same parent category are immediately adjacent in the grid. The values on the axes denote the ranges of each hierarchy level. From the plot, it can be discerned that the confusion rates are generally higher at higher hierarchy levels. Additionally, it may be noted that higher confusion rates commonly form clusters, which might suggest that categories with the same parent category often get confused. Table 5.2 lists the ten highest confusion rates for BERT_G, where Category 1 is confused with Category 2. The second column shows the unnormalized number of examples where the confusion occurs. Studying the category codes, it can be observed that in eight pairs, the categories share the same parent category; in the other two pairs, the categories are seemingly related. This suggests that, BERT_G in this particular case, is prone to confusing categories that are subject related.

Conf. Rate (%)	# Examples	Category 1	Category 2
65.7	23	Hockeytrean (RYF-QPR-HGB-BQN-TKF)	Hockeytvåan (RYF-QPR-HGB-BQN-KNN)
55.0	22	Hockeyettan Västra vår (RYF-QPR-HGB-BQN-PYW)	Hockeyettan (RYF-QPR-HGB-BQN-TQL)
40.7	11	Försvars- & säkerhetspolitik (RYF-KNI-XNS-TDS)	Försvaret (RYF-KNI-RLW-YIU)
40.0	12	Aktiehandel (RYF-IXA-CVI-MAB)	Företagsekonomi (RYF-IXA-LHV-QOI)
38.3	18	Sportarenor (RYF-QPR-ICL)	Sporter (RYF-QPR-HGB)
33.7	30	Elitserien dam (RYF-QPR-HGB-OQU-LWB)	Elitserien herr (RYF-QPR-HGB-OQU-KCB)
33.3	15	Opera (RYF-XKI-JUJ)	Teater & musikal (RYF-XKI-GJH)
33.3	11	Telenät (RYF-HPT-YGE-KEG)	Internet (RYF-HPT-YGE-EWJ)
33.3	7	Kirurgi (RYF-WUU-SUX-APJ-FCV)	Akutsjukvård (RYF-WUU-SUX-APJ-DZQ)
30.9	17	Kidnappning (RYF-BIZ-WZJ-OUB)	Hot & trakasserier (RYF-BIZ-WZJ-SVK)

Table 5.2: The ten highest confusion rates for BERT_G

Given the similarity between the most frequently confused categories above, it is relevant to reflect on how the fine-tuning dataset has been annotated. Apart from the articles being labeled by a multitude of different journalists, they are also published on different news brands. It is not unconceivable that different newsrooms interpret and utilize the category taxonomy in slightly diverging ways. Also considering the large number of categories, it is fairly likely that the same article, in theory, could receive diverging label sets from two different annotators. This could make the studied classification problem non-trivial for even a human expert. Likewise, it is possibly difficult for a classifier to learn a single correct way of labeling articles. To conclude this part of the discussion, Table 5.3 presents a handful of illustrative examples. These are test-set predictions by the best-performing classifier META_L, together with the title

and ground truth labels for each article. Categories appearing in both the ground truth and predictions are marked in bold.

Title	Ground Truth	Prediction
De ger extra språkstöd till vikarier i äldrevården: "En trygghet"	**Samhälle & välfärd** Arbetsmarknad Välfärd & bidragsfrågor Kompetensutveckling Långtids- & äldrevård	**Samhälle & välfärd** Hälsa & sjukvård Vård Äldreomsorg (Sjukvård)
Farmen-Jens från Nordmaling fortsätter stå i rampljuset: "Trodde jag skulle försvinna då Farmen var över"	**Kultur & nöje** **Medier** Underhållningsprogram Sociala medier	**Kultur & nöje** **Medier** Reality-tv
"Det är en stor investering och ytterligare en stor satsning i Skara"	**Ekonomi, näringsliv & finans** **Näringsliv** Konsumtions- & dagligvaror Detaljhandel	**Ekonomi, näringsliv & finans** **Näringsliv** Ekonomi Investeringar
MP: Ja till koldioxidbudget!	**Politik** **Miljö** **Politiska frågor** Klimatförändringar Föroreningar Bevarande av miljö **Miljöpolitik** **Kommunpolitik** Naturföroreningar Miljöfrämjande arbete	**Politik** **Miljö** **Politiska frågor** **Miljöpolitik** **Kommunpolitik**
Man frias från misstankar om våldtäktsförsök	**Brott & straff** **Rättsprocessen** **Frikännande**	**Brott & straff** Brottslighet **Rättsprocessen** Sexualbrott **Frikännande** Våldtäkt
På kolonilotterna hoppas man på kommunalt vatten	**Livsstil & fritid** Väder Hem & trädgård Väderfenomen	**Livsstil & fritid** Miljö Friluftsliv Vatten
Falun får nytt hundcafé – caféägaren Annelie: "Hunden får en egen meny"	**Ekonomi, näringsliv & finans** Näringsliv Restaurang & catering Kafé	Personligt **Ekonomi, näringsliv & finans** Husdjur
Eric Björkander förlänger – så länge blir han kvar i GIF Sundsvall	**Sport** Övergångar **Sporter** **Fotboll** Allsvenskan	**Sport** **Sporter** **Fotboll** Superettan

Table 5.3: Example predictions by META_{L}

5.2 Method

There are some methodological aspects of this thesis that are relevant to discuss. An important insight to convey is that much of the method's limitations originate from the finite amount of computational resources available for carrying out the project.

5.2.1 Methodological Considerations

As discussed in Section 5.1.7, it is plausible that the studied HMTC problem is fairly difficult, partly due to the inconsistent labeling of articles. When creating the fine-tuning dataset described in Section 3.3.2, an alternative approach could have been to impose additional constraints on the included articles. Such constraints could be to only use articles with a number of labels within a certain range, or that are labeled with categories at least down to a specific hierarchy level. This would have led to a more coherent labeling and likely reduced the complexity of the problem, resulting in better performance for all classifiers and possibly how they compare to each other. One might also expect that the coherence of the labeling would be reflected in the predictions made by the classifiers, which could be favorable in practical use cases. However, it is important to note that this approach would have reduced the dataset size and assumably the number of categories used to label at least 100 articles.

Given the inherent class imbalance in HMTC problems and the overall low recalls reported, it could have been favorable to consider alternative loss functions. For example, weighted binary cross-entropy loss could have been considered, where each category is individually weighted to balance the precision and recall. It would also have been worthwhile to explore the use of metadata features more extensively. Different scalings and combinations of the features could have been explored, as well as multiple ways of combining the metadata and textual features. Additionally, it would have been interesting to perform ablation studies on using HTrans when training local classifiers.

An effective domain specialization of BERT may be reliant on an accurate and consistent sentence segmentation of the pre-training corpus, as discussed in Section 5.1.4. In light of this, it would have been sensible to investigate alternative models for sentence segmentation, such as algorithms developed specifically for Swedish. Moreover, it could have been favorable to study the effects on downstream performance when increasing the number of additional pre-training steps. Throughout all fine-tunings of BERT and ELECTRA, no classifier started to exhibit signs of overfitting, an example of which can be seen in Table 4.1. Also considering that previous work has shown that BERT's performance on TC can improve when training for more than four epochs [3], it could have been desirable to train the classifiers for more epochs.

5.2.2 Replicability, Reliability and Validity

In an attempt to lend this thesis a high level of replicability, the employed method for obtaining all results has been thoroughly described in Chapter 3. This encompasses the complete process for composing the datasets, and for creating and evaluating the models. Additionally, the setup and tools utilized at each step have been specified in Section 3.7. Since the source code created for this work will be productionized at Bonnier News, it is not publicly available, which could hurt the replicability of the study. However, all solutions have been implemented using popular and well-documented open-source software, which should enable replication of a functionally equivalent code base. In DL research, outcomes of experiments may vary greatly depending on what frameworks and framework versions are used [16]. To address this variability, all libraries and frameworks utilized in this work, together with their corresponding versions, have been listed in Appendix B. Naturally, the

unavailability of the article data is detrimental to the replicability of this thesis. In practice, the study can only be replicated by those with access to Bonnier News' article data.

A growing concern in DL and NLP research is the reliability and reproducibility of results [16]. Consequently, several measures have been undertaken in this work to address some of the most common issues related to these concepts. The choice of hyperparameter settings can have a significant impact on the variability and effectiveness of DL models. Accordingly, all hyperparameters in this study have been carefully tuned or selected based on previous work, and are explicitly defined throughout Chapter 3. Further, the hardware on which experiments are run may influence results greatly due to, for example, different threadings and processor architectures. To this end, a single GPU model and predefined VM image have been employed for all experiments, using deterministic settings for the GPU backend library. However, since the experiments were run on a cloud platform, there is no guarantee that perfectly identical resources have been allocated to every VM instance. For ML models, randomness is an inherent feature that can influence effectiveness. To reduce the impact of randomness on data sampling and model training, a fixed random seed has consistently been used across all experiments. In addition, the effect of randomness on the final scores was investigated by repeating one experiment with multiple random seeds. The relative standard deviation of the F_1 score on the test set was 0.3%, indicating that randomness only had a moderate impact on the end results.

Due to resource constraints, some of the actions mentioned above were not possible to perform exhaustively. For instance, it would have been desirable to train and evaluate every classifier across multiple hyperparameter settings and random seeds to further increase the validity of the study. This is not unique for this thesis; much of the referenced work mentions resource constraints as the prohibiting factor for rerunning experiments. Because DL models in general, and Transformer-based models in particular, are very resource intensive, it may simply not be attainable to retrain them multiple times. Furthermore, it should be emphasized that the findings in this work are highly conditioned on the utilized data and pre-trained models. There is no guarantee that the findings hold for HMTC problems in, for example, different domains or languages.

5.2.3 Source Criticism

The references collected for this thesis have been carefully selected based on their subject relevance and scientific rigor. A majority of the references are either peer-reviewed research articles – published in journals or conference proceedings – or preprints. Because the subject area is very recent, so is the related work; more than 80% of the references are published within the last five years. As a result, much of the referenced work is not only up-to-date, but also at the forefront of NLP and TC research. For the same reason, several articles are not extensively cited, which could hurt their credibility. Referenced articles published prior to the 2010's are mostly background work that have been included here as primary sources, commonly cited in more current work. The references also encompass multiple survey papers that summarize the works from a distinct field. These are included because they structure and impose frameworks on large compilations of research, making them convenient when referencing a general trend or method. Two books have been referenced, which cover some fundamental concepts in NLP. Additionally, a few reports have been referenced, mostly on the topic of digitization and AI in the news media industry.

When referencing certain types of resources, such as software documentation or source code, there is occasionally not a formally published source to cite. In such instances, footnotes have been injected into the text, pointing to websites where the information is accessible. Evidently, there is no proper reference to published information about how KBLab created

their Swedish ELECTRAs – simply because there currently exists none. Two representatives of KBLab were contacted during this thesis work, one of whom responded, but no further information could be obtained.

5.3 The Work in a Wider Context

Research is seldomly, if ever, conducted in a contextual vacuum. More often than not, there are some broader ethical or societal aspects to consider, and this thesis is no exception.

5.3.1 Algorithmic Transparency

Algorithmic transparency in news media refers to the principle of publicly disclosing information about the algorithmic processes involved in producing a news product [22]. The idea is to provide consumers and other interested parties with a basis for assessing the underlying ideologies, values and biases that a news product is built on. As the use of algorithmic tools, including AI, increase in newsrooms, so does the demand for algorithmic transparency. This is especially true for tools that could pose ethical concerns, such as algorithmic personalization creating filter bubbles or automated news generation facilitating dissemination of misinformation [7]. If such tools malfunction, they risk having consequences on a societal level, for example, by increasing political polarization and unawareness. The risks associated with automated tagging of news text are perhaps not as obvious. However, classification systems can be utilized to generate metadata on which other algorithms operate, such as systems for content recommendation or personalized advertising. Albeit not immediately relevant for this thesis, there may in such cases arise concerns regarding, for instance, societal biases in the data and category labels used for training classifiers. For news organizations, there could exist counterincentives to algorithmic transparency, such as cost savings or maintaining a competitive advantage. Nevertheless, to remain credible, news organizations should strive for algorithmic transparency. Ultimately, this will enable consumers and other stakeholders to judge what news sources they want to trust and support.

5.3.2 Energy Costs

Something that has been reiterated throughout this work is the high resource consumption associated with contextualized language models. The required resources are often specialized hardware, such as GPUs, and are typically expensive to access and power, and naturally demand energy to function. Currently, energy is not derived from carbon-neutral sources in many locations, causing the development of DL models to not only have a monetary, but also an environmental cost. For instance, pre-training a base-size BERT model from scratch costs several thousands of dollars and emits approximately as much carbon emissions as a trans-American flight [62]. BERT is by no means an isolated phenomenon and from 2013 to 2019, the amount of computational resources used to train DL models grew by a staggering factor of 300,000 [55]. This reflects an overarching trend in DL fields such as NLP, where the most computationally-hungry models often produce the best results. As DL becomes increasingly productionized and practically applicable, it also becomes increasingly important to weigh models' absolute performances against the cost of developing and using them; that is, to address the energy efficiency of DL solutions. Some aspects of this thesis can be related to the notion of efficiency: considering the eligibility of a less complex and resource intensive alternative; reusing and repurposing pre-trained models through transfer learning; and investigating model compression techniques. However – and perhaps symptomatically – these are all fairly trivial examples and efficiency is not the main focus of this thesis. Moving forward, DL practitioners and researchers should progressively grant efficiency a higher priority in industry and academia.

6 Conclusion

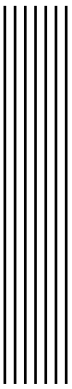
This thesis has investigated and evaluated approaches to applying pre-trained contextualized language models to news article classification. By creating news article datasets to fine-tune and evaluate Swedish versions of BERT and ELECTRA, this thesis has addressed four research questions.

Firstly, using a global and local classifier approach, maximum F_1 scores of 0.6340 and 0.6629, respectively, were obtained. For both models, the local classifier approach was superior to the global classifier approach, especially for more sparse categories. Remarkably, BERT significantly outperformed ELECTRA across all comparable experiments. Using an SVM-based baseline, a notable F_1 score of 0.6285 was obtained, thereby outperforming all ELECTRA-based classifiers. Secondly, the effect of further in-domain pre-training on classification performance was variable; ELECTRA's performance improved by 5.3%, while BERT's performance was largely unaffected. A possible explanation for the diverging observations could be a deficient sentence segmentation of BERT's pre-training data. Thirdly, metadata features, such as news brand and various text statistics, were compiled from the articles. By combining these with the original textual features, the classification performance of BERT was improved by 1%. Fourthly, quantizing models from FP32 to FP16 did not affect classification performance, while quantization to INT8 produced a maximum performance drop of 12%. Pruning up to 45% of all model weights, classification performance was unaffected. After pruning 60% of the weights, classification performance started to degrade, again with a maximum performance drop of 12%. These findings suggest that up to 50% of the information in a Transformer-based classifier can be redundant. Concludingly, the relatively strong performance of the baseline and the overall low F_1 scores are likely attributable to the large number of categories and the complexity of the classification task.

Future Work

Given the significant differences between BERT and ELECTRA, future work could focus on more comprehensive comparisons between the two models and investigate if and how their performances differ. This could be done with the Swedish, or other versions, of the models, comparing them both for TC and other downstream NLP tasks. Additionally, the overall low performance on the HMTC problem – and the relatively strong performance of the SVM

baseline – poses two interesting research directions. Future work could try to improve ensembles of more lightweight ML models, such as SVMs and decision trees, for similar HMTC tasks. This could be done by, for example, exploring different features to utilize in conjunction with SVMs, including contextual and metadata features. Alternatively, future research could focus on improving the performance of classifiers built on pre-trained contextualized language models. Possible approaches could be to use more local classifiers or to explicitly encode hierarchical information in the data samples. Furthermore, an appealing research avenue, that has not yet been extensively explored in the context of Transformer-based models, are methods for balancing the precision and recall of classifiers. Potential aspects to consider could be different loss functions and strategies for over- and undersampling data.



Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. "TensorFlow: A System for Large-Scale Machine Learning". In: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 2016, pp. 265–283.

[2] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, William L Hamilton, and Jimmy Lin. "Exploring the Limits of Simple Learners in Knowledge Distillation for Document Classification With DocBERT". In: *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP)*. 2020, pp. 72–77.

[3] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. "DocBERT: BERT for Document Classification". In: *arXiv preprint arXiv:1904.08398* (2019).

[4] Rami Aly, Steffen Remus, and Chris Biemann. "Hierarchical Multi-Label Classification of Text With Capsule Networks". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL): Student Research Workshop*. 2019, pp. 323–330.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *Proceedings of the 3rd Annual International Conference on Learning Representations (ICLR)*. 2015.

[6] Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. "Hierarchical Transfer Learning for Multi-Label Text Classification". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2019, pp. 6295–6300.

[7] Charlie Beckett. *New Powers, New Responsibilities: A Global Survey of Journalism and Artificial Intelligence*. Tech. rep. Polis, London School of Economics and Political Science, 2019.

[8] Iz Beltagy, Kyle Lo, and Arman Cohan. "SciBERT: A Pretrained Language Model for Scientific Text". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 3615–3620.

[9] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[10] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research (JMLR)* 13.2 (2012), pp. 281–305.

[11] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "An Empirical Study on Large-Scale Multi-Label Text Classification Including Few and Zero-Shot Labels". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 7503–7515.

[12] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "LEGAL-BERT: The Muppets Straight Out of Law School". In: *Findings of the Association for Computational Linguistics: EMNLP*. 2020, pp. 2898–2904.

[13] Jianpeng Cheng, Li Dong, and Mirella Lapata. "Long Short-Term Memory-Networks for Machine Reading". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2016, pp. 551–561.

[14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1724–1734.

[15] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. "ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators". In: *Proceedings of the 8th Annual International Conference on Learning Representations (ICLR)*. 2020.

[16] Matt Crane. "Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results". In: *Transactions of the Association for Computational Linguistics (TACL)* 6 (2018), pp. 241–252.

[17] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. "GAN-BERT: Generative Adversarial Learning for Robust Text Classification With a Bunch of Labeled Example". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, pp. 2114–2119.

[18] Washington Cunha, Sérgio Canuto, Felipe Viegas, Thiago Salles, Christian Gomes, Vitor Mangaravite, Elaine Resende, Thierson Rosa, Marcos André Gonçalves, and Leonardo Rocha. "Extended Pre-Processing Pipeline for Text Classification: On the Role of Meta-Feature Representations, Sparsification and Selective Sampling". In: *Information Processing & Management* 57.4 (2020).

[19] Washington Cunha, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. "On the Cost-Effectiveness of Neural and Non-Neural Approaches and Representations for Text Classification: A Comprehensive Comparative Study". In: *Information Processing & Management* 58.3 (2021).

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 2019, pp. 4171–4186.

[21] Nicholas Diakopoulos. "Computational Journalism and the Emergence of News Platforms". In: *The Routledge Companion to Digital Journalism Studies* (2017), pp. 176–184.

[22] Nicholas Diakopoulos and Michael Koliska. "Algorithmic Transparency in the News Media". In: *Digital Journalism* 5.7 (2017), pp. 809–828.

[23] Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. "CogLTX: Applying BERT to Long Texts". In: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*. 2020, pp. 12792–12804.

[24] Andrea Galassi, Marco Lippi, and Paolo Torroni. "Attention in Natural Language Processing". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[25] Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Deming Chen, Marianne Winslett, Hassan Sajjad, and Preslav Nakov. "Compressing Large-Scale Transformer-Based Models: A Case Study on BERT". In: *arXiv preprint arXiv:2002.11985* (2020).

[26] Santiago González-Carvajal and Eduardo C. Garrido-Merchán. "Comparing BERT Against Traditional Machine Learning Text Classification". In: *arXiv preprint arXiv:2005.13012* (2020).

[27] Mitchell A Gordon, Kevin Duh, and Nicholas Andrews. "Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning". In: *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP)*. 2020, pp. 143–155.

[28] Daniel Holmer and Arne Jönsson. *Comparing the Performance of Various Swedish BERT Models for Classification*. Tech. rep. Linköping University, 2020.

[29] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. "TinyBERT: Distilling BERT for Natural Language Understanding". In: *Findings of the Association for Computational Linguistics: EMNLP*. 2020, pp. 4163–4174.

[30] Thorsten Joachims. "Text Categorization With Support Vector Machines: Learning With Many Relevant Features". In: *Proceedings of the 10th European Conference on Machine Learning (ECML)*. 1998, pp. 137–142.

[31] Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. "I-BERT: Integer-Only BERT Quantization". In: *arXiv preprint arXiv:2101.01321* (2021).

[32] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *Proceedings of the 8th Annual International Conference on Learning Representations (ICLR)*. 2020.

[33] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[34] Qian Li, Hao Peng, Jianxin Li, Congyin Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. "A Text Classification Survey: From Shallow to Deep Learning". In: *arXiv preprint arXiv:2008.00364* (2020).

[35] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. "Train Big, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 2020, pp. 5958–5968.

[36] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[37] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[38] Zhibin Lu, Pan Du, and Jian-Yun Nie. "VGCN-BERT: Augmenting BERT With Graph Embedding for Text Classification". In: *Proceedings of the 42nd European Conference on Information Retrieval (ECIR)*. 2020, pp. 369–382.

[39] Martin Malmsten, Love Börjeson, and Chris Haffenden. "Playing With Words at the National Library of Sweden – Making a Swedish BERT". In: *arXiv preprint arXiv:2007.01658* (2020).

[40] Aditya Malte and Pratik Ratadiya. "Evolution of Transfer Learning in Natural Language Processing". In: *arXiv preprint arXiv:1910.07370* (2019).

[41] Bertin Martens, Luis Aguiar, Estrella Gomez-Herrera, and Frank Mueller-Langer. *The Digital Transformation of News Media and the Rise of Disinformation and Fake News*. Tech. rep. Joint Research Centre (JRC), 2018.

[42] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. "CamemBERT: A Tasty French Language Model". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, pp. 7203–7219.

[43] Nick Newman, Richard Fletcher, Anne Schulz, Sigme Andı, and Rasmus Kleis Nielsen. *Reuters Institute Digital News Report 2020*. Tech. rep. Reuters Institute for the Study of Journalism, 2020.

[44] Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp. "Enriching BERT with Knowledge Graph Embeddings for Document Classification". In: *Proceedings of the GermEval 2019 Workshop*. 2019.

[45] Michela Paganini and Jessica Forde. "Streamlining Tensor and Network Pruning in PyTorch". In: *arXiv preprint arXiv:2004.13770* (2020).

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *arXiv preprint arXiv:1912.01703* (2019).

[47] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research (JMLR)* 12 (2011), pp. 2825–2830.

[48] Telmo Pires, Eva Schlinger, and Dan Garrette. "How Multilingual Is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2019, pp. 4996–5001.

[49] Qi Qin, Wenpeng Hu, and Bing Liu. "Feature Projection for Improved Text Classification". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, pp. 8161–8171.

[50] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. *Improving Language Understanding by Generative Pre-Training*. Tech. rep. OpenAI, 2018.

[51] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[52] Steffen Remus, Rami Aly, and Chris Biemann. "GermEval 2019 Task 1: Hierarchical Classification of Blurbs". In: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*. 2019, pp. 280–292.

[53] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter". In: *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC^2)*. 2019.

[54] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[55] Roy Schwartz, Jesse Dodge, N.A. Smith, and Oren Etzioni. “Green AI”. In: *Communications of the ACM* 63 (2020), pp. 54–63.

[56] Fabrizio Sebastiani. “Machine Learning in Automated Text Categorization”. In: *ACM Computing Surveys* 34.1 (2002), pp. 1–47.

[57] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. “Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT”. In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. 2020, pp. 8815–8821.

[58] Wen Shen, Zhihua Wei, Qianwen Li, Hongyun Zhang, and Duoqian Miao. “Three-Way Decisions Based Blocking Reduction Models in Hierarchical Classification”. In: *Information Sciences* 523 (2020), pp. 63–76.

[59] Carlos Silla and Alex Freitas. “A Survey of Hierarchical Classification Across Different Application Domains”. In: *Data Mining and Knowledge Discovery* 22 (2011), pp. 31–72.

[60] Mohammad S. Sorower. *A Literature Survey on Algorithms for Multi-Label Learning*. Tech. rep. Oregon State University, 2010.

[61] Milan Straka and Jana Straková. “Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe”. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. 2017, pp. 88–99.

[62] Emma Strubell, Ananya Ganesh, and Andrew McCallum. “Energy and Policy Considerations for Deep Learning in NLP”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2019, pp. 3645–3650.

[63] Aixin Sun, Ee-Peng Lim, and Ying Liu. “On Strategies for Imbalanced Text Classification Using SVM: A Comparative Study”. In: *Decision Support Systems* 48.1 (2009), pp. 191–201.

[64] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. “How to Fine-Tune BERT for Text Classification?” In: *Proceedings of the 18th China National Conference on Chinese Computational Linguistics (CCL)*. 2019, pp. 194–206.

[65] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. “Patient Knowledge Distillation for BERT Model Compression”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 4323–4332.

[66] Grigorios Tsoumakas and Ioannis Katakis. “Multi-Label Classification: An Overview”. In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007).

[67] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Well-Read Students Learn Better: On the Importance of Pre-Training Compact Models”. In: *arXiv preprint arXiv:1908.08962* (2019).

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need”. In: *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 5998–6008.

[69] Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. “Multilingual Is Not Enough: BERT for Finnish”. In: *arXiv preprint arXiv:1912.07076* (2019).

[70] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. “BERTje: A Dutch BERT Model”. In: *arXiv preprint arXiv:1912.09582* (2019).

[71] Erik Wiener, Jan O. Pedersen, and Andreas S. Weigend. “A Neural Network Approach to Topic Spotting”. In: *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*. 1995, pp. 317–332.

[72] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*. 2020, pp. 38–45.

[73] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *arXiv preprint arXiv:1609.08144* (2016).

[74] Yiming Yang. "A Study on Thresholding Strategies for Text Categorization". In: *Proceedings of the 24th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*. 2001, pp. 137–145.

[75] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[76] Shanshan Yu, Jindian Su, and Da Luo. "Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge". In: *IEEE Access* 7 (2019), pp. 176600–176612.

[77] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. "Q8BERT: Quantized 8Bit BERT". In: *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC^2)*. 2019.

[78] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. "Big Bird: Transformers for Longer Sequences". In: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*. 2020, pp. 17283–17297.

[79] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76.

[80] Quan Zou, Sifa Xie, Ziyu Lin, Meihong Wu, and Ying Ju. "Finding the Best Classification Threshold in Imbalanced Classification". In: *Big Data Research* 5 (2016), pp. 2–8.

Appendix

A Dataset Statistics

If not stated otherwise, the statistics displayed below are on the fine-tuning dataset. Note that the vertical axis is occasionally logarithmically scaled.

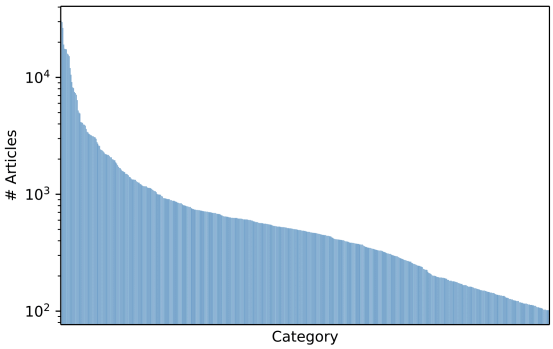


Figure A.1: Category distribution

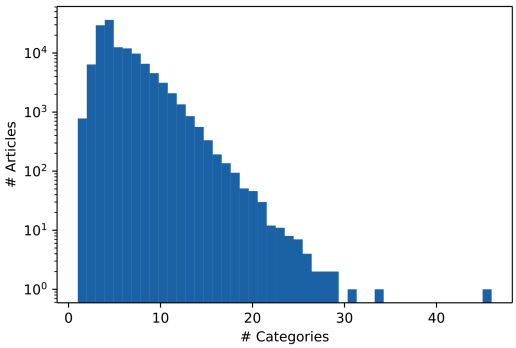


Figure A.2: Number of categories per article

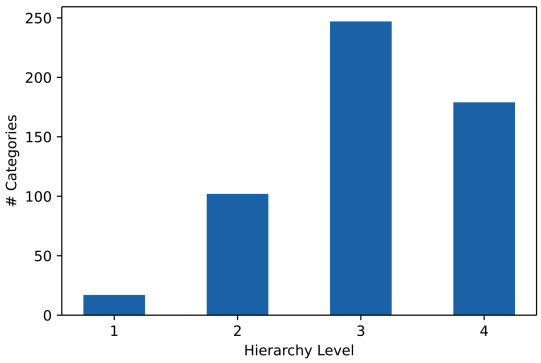


Figure A.3: Number of categories per level

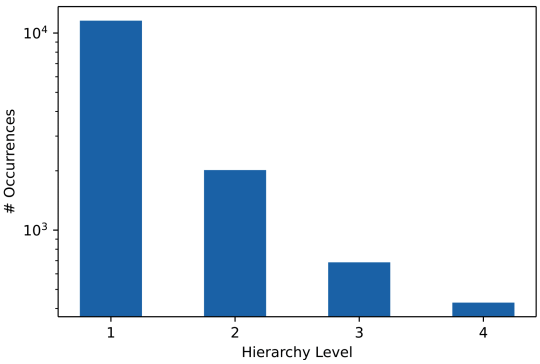


Figure A.4: Mean number of category occurrences per level

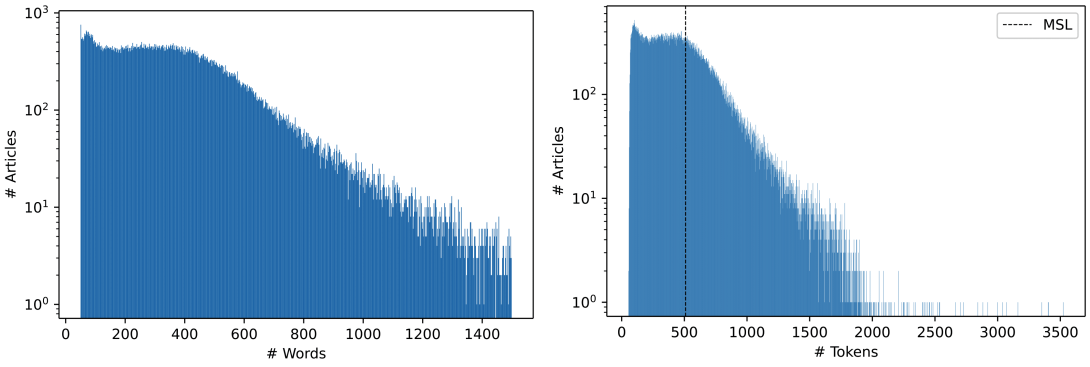


Figure A.5: Article length

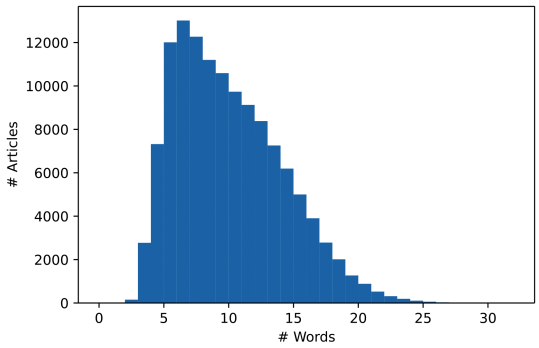


Figure A.6: Title length

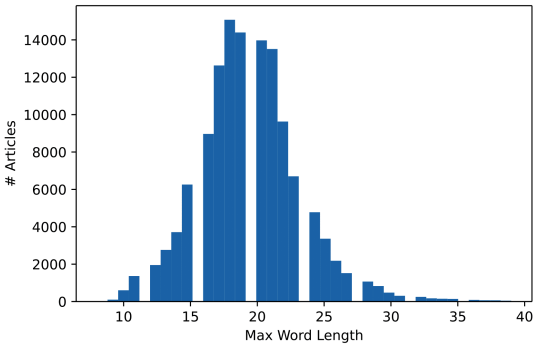


Figure A.7: Max word length

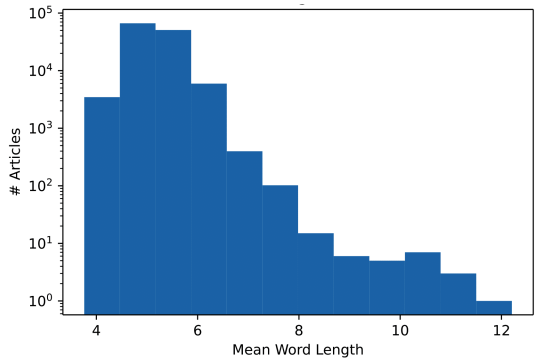


Figure A.8: Mean word length

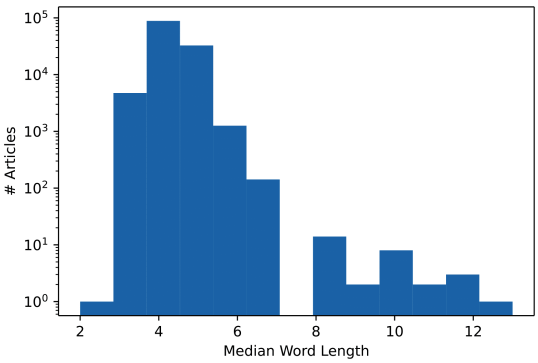


Figure A.9: Median word length

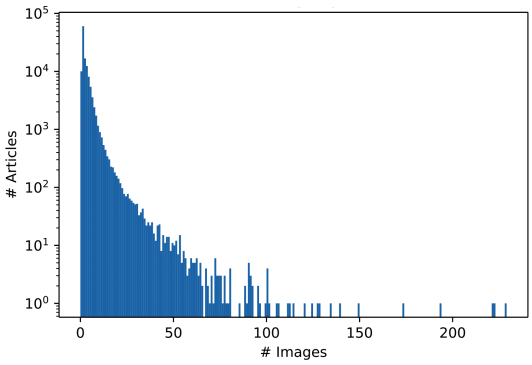


Figure A.10: Number of images per article

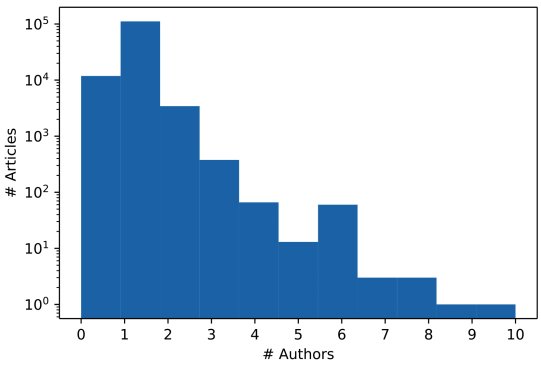


Figure A.11: Number of authors per article

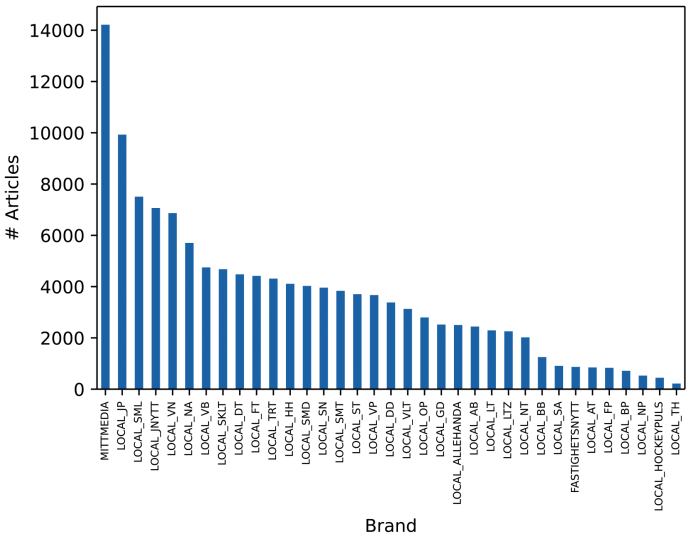


Figure A.12: Brand distribution

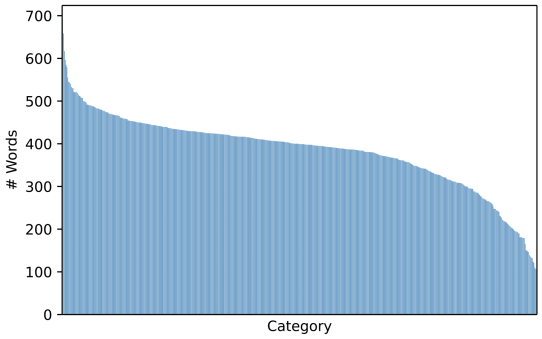


Figure A.13: Mean article length per category

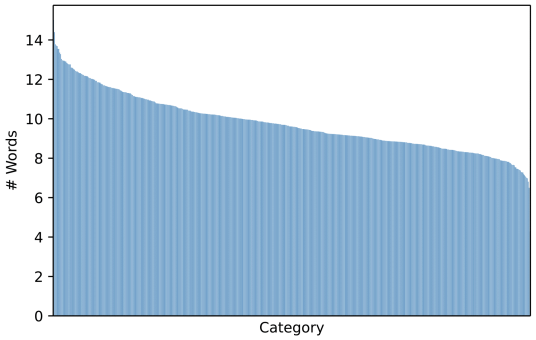


Figure A.14: Mean title length per category

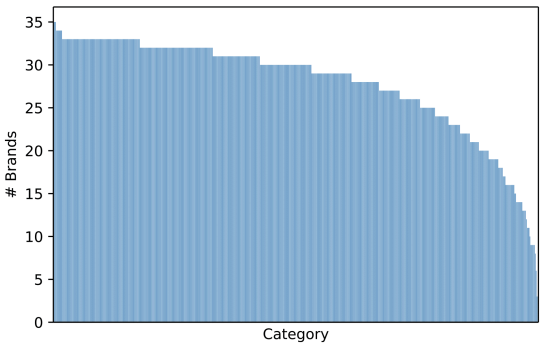


Figure A.15: Mean number of brands per category

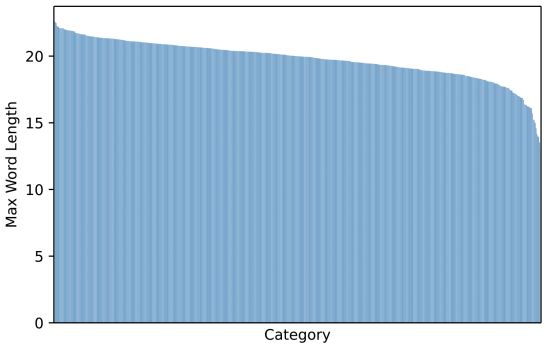


Figure A.16: Mean of max word length per category

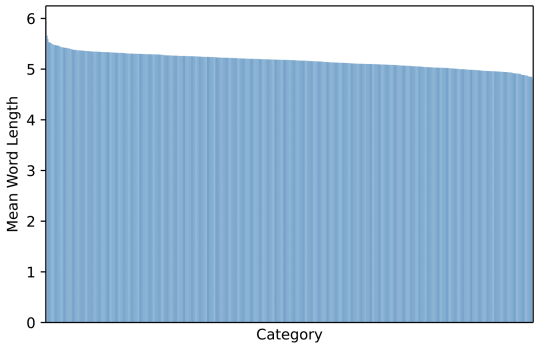


Figure A.17: Mean of mean word length per category

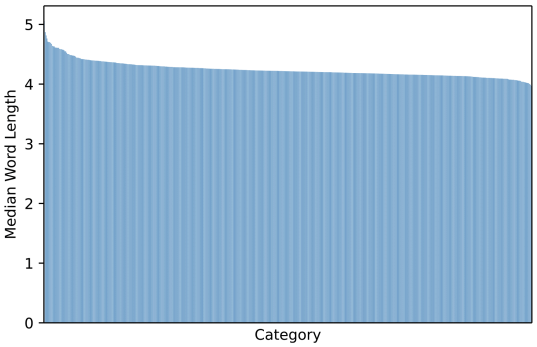


Figure A.18: Mean of median word length per category

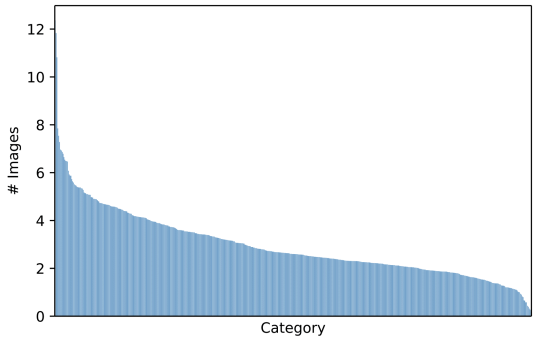


Figure A.19: Mean number of images per category

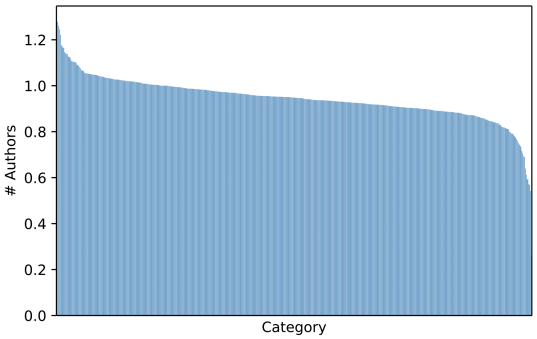


Figure A.20: Mean number of authors per category

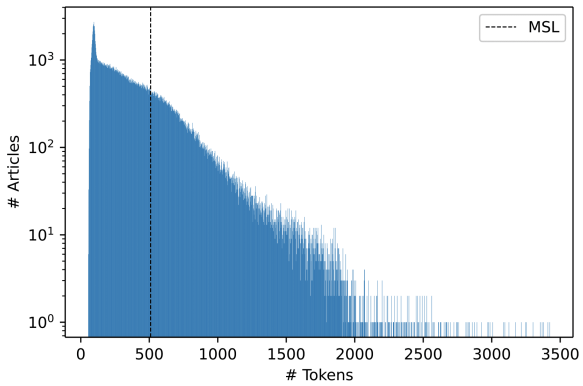


Figure A.21: Article length, pre-training dataset

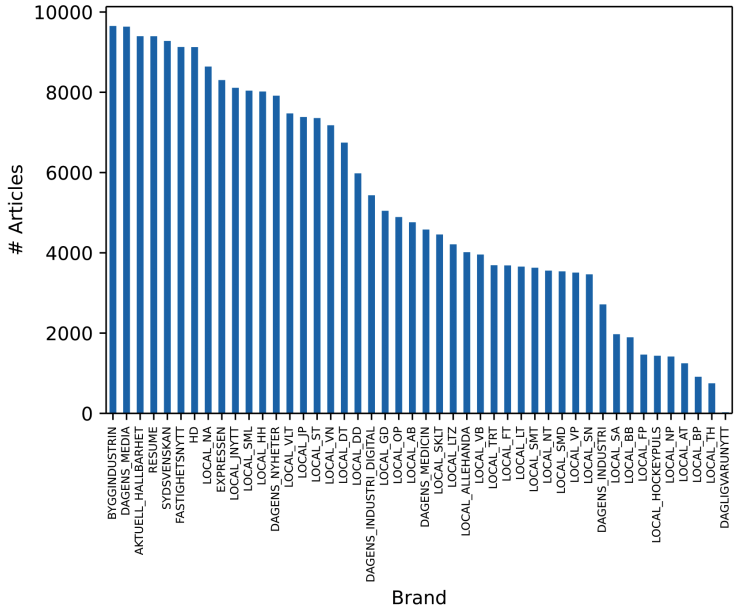


Figure A.22: Brand distribution, pre-training dataset

B Libraries

Table B.1 lists the libraries and frameworks used in this thesis together with their versions.

Library	Version
beautifulsoup4	4.9.3
emoji	1.2.0
numpy	1.20.1
pandas	1.2.3
scikit_learn	0.24.1
snapy	1.0.2
spacy_udpipe	0.3.2
torch	1.7.1
transformers	4.3.3

Table B.1: Library versions

C Category Taxonomy

In Table C.2 below, the categories in the utilized subset of the Category Tree for Swedish Local News are shown. Each category is listed together with its corresponding code, hierarchy level and, if available, description. The rows are grouped according to level and parent category.

Category	Code	Level	Description
Olyckor & katastrofer	RYF-AKM	1	Händelser som resulterar i skador på levande varelser eller skador på egendom
Brott & straff	RYF-BIZ	1	-
Personligt	RYF-CUW	1	Artiklar om individer, grupper, djur, växter eller andra föremål med fokus på känslomässiga aspekter
Vetenskap & teknologi	RYF-EOI	1	-
Samhälle & välfärd	RYF-HPT	1	Artiklar som rör samhällsfrågor som till exempel välfärden, olika människogrupper och sociala problem
Religion & tro	RYF-ITV	1	-
Ekonomi, näringsliv & finans	RYF-IXA	1	-
Politik	RYF-KNI	1	-
Sport	RYF-QPR	1	Sport som huvudämne, både specifika sporter eller sportrelaterade ämnen som dopning och idrottsaffärer
Livsstil & fritid	RYF-TKT	1	-
Miljö	RYF-VHD	1	-
Väder	RYF-WHZ	1	-
Hälsa & sjukvård	RYF-WUU	1	-
Konflikter, krig & terrorism	RYF-WXT	1	-
Kultur & nöje	RYF-XKI	1	-
Skola & utbildning	RYF-YDR	1	-
Arbetsmarknad	RYF-ZEI	1	Sociala aspekter, lagar, regler och förhållanden som påverkar sysselsättning och arbetslöshet
Brand	RYF-AKM-AUE	2	Alla former av skadliga bränder. Till exempel villabränder. Ej majbrasa eller liknande
Olyckor	RYF-AKM-QGJ	2	Alla typer av olyckor, till exempel trafikolyckor, arbetsplatsolyckor, utsläpp eller explosioner
Katastrof	RYF-AKM-TGS	2	Allvarlig händelse där samhället inte står rustat att ta hand om konsekvenserna. Till exempel svält eller naturkatastrofer
Beredskapsplanering	RYF-AKM-TYE	2	Planering för åtgärder för att ta itu med plötsliga, oförutsedda händelser
Polisiärt arbete	RYF-BIZ-GCG	2	-
Brottslighet	RYF-BIZ-WZJ	2	-
Rättsprocessen	RYF-BIZ-XSV	2	-
Husdjur	RYF-CUW-AEJ	2	Känslomässiga berättelser om djur, eller om djurs roll för människan
Familjefrågor	RYF-CUW-GIV	2	Frågor som rör familjen som institution, till exempel äktenskap, adoption och skilsmässor
Prestationer	RYF-CUW-IAA	2	Artiklar om en prestation, att man åstadkommit något speciellt eller uppnått milstole till exempel
Växter	RYF-CUW-JCL	2	Artiklar om växter med fokus på känslomässiga aspekter
Ceremonier & händelser	RYF-CUW-LGT	2	Artiklar om speciella tillfällen, ceremonier och händelser
Högtider	RYF-CUW-XVL	2	Artiklar som rör kalendariska högtider
Naturvetenskap	RYF-EOI-DLM	2	-
Akademisk forskning	RYF-EOI-DOE	2	-

Biomedicinsk vetenskap	RYF-EOI-FZS	2	-
Teknik & ingenjörskonst	RYF-EOI-GJM	2	-
Samhällsvetenskaper	RYF-EOI-GLR	2	-
Demografi	RYF-HPT-ECD	2	Vetenskapen om en befolknings fördelning, storlek och sammansättning
Människogrupper	RYF-HPT-PFF	2	Frågor om specifika grupperingar av liknande människor
Moraliska frågor & tabubelagda ämnen	RYF-HPT-PPR	2	Frågor som rör moraliska frågor, eller frågor som är svåra att prata om
Sociala problem	RYF-HPT-THK	2	Olika typer av problem som kännetecknas av de går ut över en persons omgivning
Diskriminering	RYF-HPT-XAO	2	Att göra skillnad på människor på grund av något som inte bygger på meriter eller talanger
Välfärd & bidragsfrågor	RYF-HPT-XRQ	2	Frågor som rör välfärdssamhället och den hjälp samhället kan ge människor som behöver mat, boende med mera
Infrastruktur	RYF-HPT-YGE	2	-
Samhällen	RYF-HPT-ZSS	2	Grupp av människor som hålls samman av geografiska, administrativa eller sociala relationer
Religiösa byggnader	RYF-ITV-IQC	2	Anläggning där en grupp kan utföra sina religiösa riter, till exempel moské, kyrka, hus eller tält
Religiösa åskådningar	RYF-ITV-LPR	2	-
Religiösa händelser	RYF-ITV-QAK	2	Nyheter om en religiös händelse, men inte en festival eller semester
Börser & handelsmarknader	RYF-IXA-CVI	2	-
Näringsliv	RYF-IXA-KEV	2	-
Företagande	RYF-IXA-LHV	2	-
Ekonomi	RYF-IXA-QED	2	Artikel i en allmän karaktär om handel och ekonomi
Val	RYF-KNI-MRR	2	-
Politiska processen	RYF-KNI-OHT	2	-
Myndigheter	RYF-KNI-RLW	2	Organisationer som utövar styret i ett land
Internationella relationer	RYF-KNI-SQH	2	-
Politiska frågor	RYF-KNI-XNS	2	-
Grundlagar	RYF-KNI-ZIS	2	Använd för diskussioner om politiska debatter som rör Sveriges grundläggande lagar och rättigheter
Övergångar	RYF-QPR-FMU	2	Nyheter om spelare som byter lag eller byter klubb, eller rykten om detsamma
Sporter	RYF-QPR-HGB	2	Olika typer av sporter och även grenar inom sporter
Sportarenor	RYF-QPR-ICL	2	Nyheter som berör arenor och andra faciliteter där sport utövas
Hem & trädgård	RYF-TKT-AGB	2	-
Spel & dobbel	RYF-TKT-BIY	2	-
Villa	RYF-TKT-EPN	2	-
Friluftsliv	RYF-TKT-ESA	2	Samlingsbegrepp för fritidsaktiviteter som genomförs utomhus
Mat & dryck	RYF-TKT-ESN	2	-
Fiske	RYF-TKT-FGO	2	-
Mode & styling	RYF-TKT-GSR	2	-
Hantverk	RYF-TKT-HFA	2	-
Föreningsliv	RYF-TKT-INK	2	-

Fritidshus	RYF-TKT-MIZ	2	-
Bostadsrätt	RYF-TKT-MNX	2	-
Motor	RYF-TKT-MRA	2	-
Semester	RYF-TKT-PEJ	2	-
Jakt	RYF-TKT-PYI	2	-
Båtliv	RYF-TKT-QWJ	2	-
Ridning	RYF-TKT-RRC	2	-
Cykelentusiasm	RYF-TKT-SPE	2	-
Spel	RYF-TKT-VVD	2	-
Motion & hälsa	RYF-TKT-YIG	2	-
Resa	RYF-TKT-ZWD	2	-
Klimatförändringar	RYF-VHD-GUZ	2	-
Föroreningar	RYF-VHD-IFE	2	-
Ekologi	RYF-VHD-OAY	2	-
Vatten	RYF-VHD-QTT	2	Miljöfrågor om bland annat hav, sjöar, vattendrag och reservoarer, samt isar, glaciärer och nederbörd
Bevarande av miljö	RYF-VHD-ZKU	2	Bevarande av ödemarker, flora och fauna, inklusive arter som hotas av utrotning
Väderfenomen	RYF-WHZ-LAW	2	Särskilda väderförhållanden såsom halofenomen och tromber
Vädervarningar	RYF-WHZ-MKN	2	Varningar till den allmänna befolkningen om kraftigt väder
Väderprognoser	RYF-WHZ-XTD	2	Förutsägelse av vädret i framtiden antingen kort eller lång sikt
Sjukdomar & tillstånd	RYF-WUU-JZK	2	-
Behandlingar	RYF-WUU-QNB	2	-
Vård	RYF-WUU-SUX	2	-
Oroligheter	RYF-WXT-BND	2	Missnöje bland befolkningen, vilket framgår av till exempel strejker, demonstrationer eller sabotage
Beväpnade konflikter	RYF-WXT-CRN	2	Tvister mellan motsatta grupper som innebär användning av vapen
Medier	RYF-XKI-BUS	2	-
Kulturhistoria	RYF-XKI-DEG	2	Används för historia om kultur. Till exempel återblick på kulturprofils arbete, hur en musikgenre uppstod
Musik	RYF-XKI-FEY	2	-
Underhållningsevenemang	RYF-XKI-FXL	2	Till exempel konserter, spelturneringar
Teater & musikal	RYF-XKI-GJH	2	-
Uteliv	RYF-XKI-HLO	2	Till exempel nyheter om nattklubbar, krogar eller vimmel
Seder & traditioner	RYF-XKI-IHA	2	Till exempel diskussioner om när man skall slänga ut julgranen, hur man bör hälsa på främlingar
Kulturell fest eller händelse	RYF-XKI-ISL	2	Alla typer av händelser med en viss kulturell bakgrund, inte nödvändigtvis knuten till en fast tillfälle eller datum
Arkitektur	RYF-XKI-IUV	2	Används till artiklar om arkitektur. Till exempel arkitekturtävlingar, åsikter om byggnaders utseende
Film	RYF-XKI-JIJ	2	-
Opera	RYF-XKI-JUJ	2	-
Barnkultur	RYF-XKI-KKX	2	-
Visuella konstformer	RYF-XKI-LNE	2	-
Språk	RYF-XKI-PGP	2	Använd för reportage om språk. Till exempel att kommunen stödjer ett nytt språk, språkhistoria

Bibliotek & museum	RYF-XKI-SFU	2	-
Dans	RYF-XKI-WEG	2	-
Kulturarv	RYF-XKI-YBJ	2	Till exempel Höga Kustenområdet, fornlämningar
Litteratur	RYF-XKI-YFJ	2	-
Lärarkåren	RYF-YDR-CXG	2	-
Skolsystemet	RYF-YDR-MHH	2	-
Friskola	RYF-YDR-MQP	2	En skola som inte drivs av offentliga sektorn, men som huvudsakligen finansieras av skattemedel
Kommunal skola	RYF-YDR-UER	2	En kommunal skola är öppen för alla och ingår i det offentliga skolväsendet i Sverige
Undervisning & lärande	RYF-YDR-WXH	2	-
Anställningsförhållanden	RYF-ZEI-CLV	2	De regler och lagar som reglerar en anställning, till exempel MBL
Arbetslöshet	RYF-ZEI-QWT	2	-
Arbetslagstiftningen	RYF-ZEI-RYD	2	Lagar som styr anställningsförhållanden, arbetsrätt, arbetsmiljö, arbetstagarinflytande och arbetsmarknadsreglering
Kompetensutveckling	RYF-ZEI-YYF	2	Kompletterande utbildning för att förbättra nuvarande kompetens
Fordonsbrand	RYF-AKM-AUE-CYE	3	-
Industribrand	RYF-AKM-AUE-GXF	3	-
Brand i byggnad	RYF-AKM-AUE-MPZ	3	-
Skogsbrand	RYF-AKM-AUE-MRU	3	-
Gräsbrand	RYF-AKM-AUE-ZZI	3	-
Viltolycka	RYF-AKM-QGJ-KRH	3	-
Trafikolycka	RYF-AKM-QGJ-KUV	3	-
Arbetsplatsolycka	RYF-AKM-QGJ-SMW	3	-
Drunkning	RYF-AKM-QGJ-SRF	3	-
Sjöfartsolycka	RYF-AKM-QGJ-TBS	3	-
Explosionsolycka	RYF-AKM-QGJ-TMT	3	-
Vållande till annans död	RYF-BIZ-WZJ-COP	3	-
Djurskyddsbrott	RYF-BIZ-WZJ-DCA	3	-
Trafikbrott	RYF-BIZ-WZJ-ELD	3	-
Stölder & inbrott	RYF-BIZ-WZJ-GNO	3	-
Smugglingsbrott	RYF-BIZ-WZJ-III	3	-
Allmänfarliga brott	RYF-BIZ-WZJ-IQH	3	-
Korruption	RYF-BIZ-WZJ-JDU	3	Person som missbrukar offentlig makt för privat vinning
Häleri	RYF-BIZ-WZJ-KLS	3	-
Våldsbrott	RYF-BIZ-WZJ-KVS	3	-
Vapenbrott	RYF-BIZ-WZJ-NVZ	3	-
Miljöbrott	RYF-BIZ-WZJ-OFE	3	-
Kidnappning	RYF-BIZ-WZJ-OUB	3	-
Organiserad brottslighet	RYF-BIZ-WZJ-PQY	3	Brott som begås av ligor eller kriminella grupper
Bedrägerier & ekobrott	RYF-BIZ-WZJ-RPI	3	-
Narkotikabrott	RYF-BIZ-WZJ-RXJ	3	Alla typer av brott som rör narkotika
Hot & trakasserier	RYF-BIZ-WZJ-SVK	3	-
Skadegörelse	RYF-BIZ-WZJ-TZC	3	-
Sexualbrott	RYF-BIZ-WZJ-XEY	3	-
Straff	RYF-BIZ-XSV-BII	3	-
Förundersökning	RYF-BIZ-XSV-JIV	3	-
Straffrätt	RYF-BIZ-XSV-JTA	3	-
Förvaltningsrätt	RYF-BIZ-XSV-JUZ	3	Handlägger mål som främst rör tvister mellan enskilda personer och myndigheter
Frikännande	RYF-BIZ-XSV-RSI	3	-
Övriga rättsärenden	RYF-BIZ-XSV-XBV	3	Juridiska frågor som regleras utanför domstol

Frihetsberövande	RYF-BIZ-XSV-XDC	3	Gripande, anhållan, häktning eller liknande
Priser & utmärkelser	RYF-CUW-IAA-JLY	3	Inriktat på om man fått pris eller utmärkselse
Födelsedag	RYF-CUW-LGT-CIG	3	-
Jubilar	RYF-CUW-LGT-FPF	3	-
Bröllop	RYF-CUW-LGT-NLW	3	-
Begravning	RYF-CUW-LGT-NSR	3	-
Dödsfall	RYF-CUW-LGT-PLA	3	-
Studentfirande	RYF-CUW-LGT-WOF	3	-
Jul	RYF-CUW-XVL-AJJ	3	-
Nationaldagen	RYF-CUW-XVL-DIS	3	-
Valborg	RYF-CUW-XVL-EPD	3	-
Lucia	RYF-CUW-XVL-FOK	3	-
Påsk	RYF-CUW-XVL-IEM	3	-
Midsommar	RYF-CUW-XVL-JZT	3	-
Nyår	RYF-CUW-XVL-PKH	3	-
Biologi	RYF-EOI-DLM-FKB	3	-
Medicinsk forskning	RYF-EOI-DOE-QRZ	3	-
Veterinärvetenskap	RYF-EOI-FZS-RFH	3	-
Psykologi	RYF-EOI-GLR-FWR	3	-
Historia	RYF-EOI-GLR-JHA	3	-
Juridik	RYF-EOI-GLR-KDA	3	-
Arkeologi	RYF-EOI-GLR-MVZ	3	-
Befolkningsutveckling	RYF-HPT-ECD-NOG	3	Artiklar som rör befolkningens utveckling över tid
Barn	RYF-HPT-PFF-EUU	3	-
Pensionärer	RYF-HPT-PFF-KWE	3	-
Nationell eller etnisk minoritet	RYF-HPT-PFF-QXE	3	-
HBTQ-personer	RYF-HPT-PFF-YBM	3	-
Tonåringar	RYF-HPT-PFF-ZFA	3	-
Döden	RYF-HPT-PPR-LQJ	3	-
Fattigdom	RYF-HPT-THK-IIG	3	-
Hemlöshet	RYF-HPT-THK-KAW	3	-
Utanförskap	RYF-HPT-THK-KFS	3	Personer som är arbetsoförmögna, antingen fysiskt, känslomässigt eller psykiskt
Prostitution	RYF-HPT-THK-OIW	3	-
Missbruk	RYF-HPT-THK-RYE	3	-
Ungdomsbrottslighet	RYF-HPT-THK-TPB	3	-
Könsdiskriminering	RYF-HPT-XAO-JSW	3	-
Rasism	RYF-HPT-XAO-OHC	3	-
Långtids- & äldrevård	RYF-HPT-XRQ-DXC	3	Omfattande sjukvård på grund av allvarlig sjukdom eller funktionshinder till exempel på grund av ålder
Skolskjutsar	RYF-HPT-XRQ-HQH	3	-
Bidrag	RYF-HPT-XRQ-KWB	3	Pengar individer kan få av myndigheter för att täcka basala behov
Välgörenhet	RYF-HPT-XRQ-MLN	3	Att skänka pengar till organisation eller person för en specifik orsak
Barnomsorg	RYF-HPT-XRQ-UFA	3	-
Socialt skyddsnät	RYF-HPT-XRQ-YNP	3	Frågor som rör personers sociala nätverk som kan verka stöttande för en person med sociala problem
Kollektivtrafik	RYF-HPT-YGE-AAD	3	-
Internet	RYF-HPT-YGE-EWJ	3	-
Parkering	RYF-HPT-YGE-FAZ	3	-
Vattennät	RYF-HPT-YGE-GCZ	3	-
Trafikpåverkan	RYF-HPT-YGE-GUI	3	-
Telenät	RYF-HPT-YGE-KEG	3	-
Tågnät	RYF-HPT-YGE-MOZ	3	-

Vägnät	RYF-HPT-YGE-USH	3	-
Elnät	RYF-HPT-YGE-WRY	3	-
Stadsmiljö	RYF-HPT-ZSS-HAB	3	-
Samhällsplanering	RYF-HPT-ZSS-WTL	3	-
Kristendom	RYF-ITV-LPR-BCK	3	-
Religiösa ritualer	RYF-ITV-QAK-VBF	3	Etablerade religiösa ritualer som till exempel gudstjänst, begravning och vigsel
Råvarumarknad	RYF-IXA-CVI-FUJ	3	-
Aktiehandel	RYF-IXA-CVI-MAB	3	-
IT & datateknik	RYF-IXA-KEV-BEL	3	-
Gruvnäring	RYF-IXA-KEV-CKN	3	Produktion, prospektering och raffinering av malm i metaller, ofta från gruvor
Avfallshantering & bekämpning av föroreningar	RYF-IXA-KEV-DOS	3	Verksamheter inom avfallshantering och begränsning av föroreningar
Fastighets- & byggnadsindustri	RYF-IXA-KEV-OTN	3	-
Restaurang & catering	RYF-IXA-KEV-RFJ	3	-
Konsumtions- & dagligvaror	RYF-IXA-KEV-RQL	3	-
Reklam & PR	RYF-IXA-KEV-SXR	3	-
Energi, naturtillgångar & elproduktion	RYF-IXA-KEV-UBU	3	-
Privat- & företagstjänster	RYF-IXA-KEV-VKR	3	-
Processade tjänster & produkter	RYF-IXA-KEV-VRA	3	Verksamheten för att omvandla råvaror till användbara produkter
Naturbruk	RYF-IXA-KEV-VSA	3	-
Transporter	RYF-IXA-KEV-VXX	3	Medlen för att komma från en plats till en annan utan att gå
Tillverkning & konstruktion	RYF-IXA-KEV-WHS	3	-
Kemisk industri	RYF-IXA-KEV-XZV	3	-
Turism & besöksnäring	RYF-IXA-KEV-YII	3	-
Egenföretagande	RYF-IXA-LHV-DFG	3	-
Nyetableringar	RYF-IXA-LHV-FFA	3	Alla typer av etableringar av nya företag, oavsett bransch
Bolagsstyre	RYF-IXA-LHV-GNH	3	-
Företagsekonomi	RYF-IXA-LHV-QOI	3	-
Investeringar	RYF-IXA-QED-CPE	3	Använd för investeringar i råvaror, värdepapper, valutor eller andra spekulativa instrument
Pengar & penningpolitik	RYF-IXA-QED-CQI	3	-
Pension	RYF-IXA-QED-HZF	3	Frågor som rör pensionen, till exempel pensionsålder, förtidspension med mera
Krediter & skulder	RYF-IXA-QED-KZN	3	-
Budgetar & budgetering	RYF-IXA-QED-LIJ	3	-
Fastighetsaffärer	RYF-IXA-QED-QAS	3	Köp och försäljning av fastigheter av alla typer
Export	RYF-IXA-QED-RHY	3	-
Privatekonomi	RYF-IXA-QED-UGD	3	-
Ekonomisk tillväxt	RYF-IXA-QED-VZV	3	-
Bistånd	RYF-IXA-QED-WGO	3	-
Rösta i val	RYF-KNI-MRR-EIU	3	Använd för nyheter som rör själva röstandet. Till exempel att färre taktikröstar eller om en röstlokal är hotad
Kommunval	RYF-KNI-MRR-NUJ	3	-
Medborgarförslag	RYF-KNI-MRR-URH	3	Politiska förslag som inte grundas på styrets egna utredningar
Folkomröstning	RYF-KNI-MRR-YYH	3	Använd för politiska beslut som fattas av en direkt omröstning hos samtliga röstberättigade medborgare
Politiska debatter	RYF-KNI-OHT-ACB	3	-

Politiskt system	RYF-KNI-OHT-PIO	3	System utformad för att ge order till regeringen
Myndighetsarbete	RYF-KNI-RLW-HER	3	-
Försvaret	RYF-KNI-RLW-YIU	3	-
Flyktingfrågan globalt	RYF-KNI-SQH-DKK	3	Människor som söker skydd i ett annat land för att undslippa förföljelse och misär
Infrastrukturfrågor	RYF-KNI-XNS-CFE	3	Till exempel anslag till vägar, internetanslutning eller elnät
Kulturpolitik	RYF-KNI-XNS-CHY	3	Till exempel anslag till kulturella föreningar och sällskap
Miljöpolitik	RYF-KNI-XNS-ELZ	3	Till exempel investeringar i förnybara resurser och skatt på utsläpp
Regions- & landstingsfrågor	RYF-KNI-XNS-FQO	3	Används för artiklar som berör regioners och landstings styre
Kommunpolitik	RYF-KNI-XNS-FYA	3	-
Skattepolitik	RYF-KNI-XNS-HIC	3	-
Flyktingpolitik	RYF-KNI-XNS-JVN	3	-
Sjukvårdspolitik	RYF-KNI-XNS-KVO	3	-
Samhällspolitik	RYF-KNI-XNS-SWE	3	-
Försvars- & säkerhetspolitik	RYF-KNI-XNS-TDS	3	Regeringens politik för att skydda nationen och gränserna
Regleringar	RYF-KNI-XNS-TVM	3	-
Ekonomisk politik	RYF-KNI-XNS-WZY	3	-
Landsbygdspolitik	RYF-KNI-XNS-YQW	3	-
Yttrandefrihet & tryckfrihet	RYF-KNI-ZIS-IUN	3	Personers rätt att kommunicera och uttrycka åsikter och idéer
Mänskliga rättigheter	RYF-KNI-ZIS-OPA	3	Universiella rättigheter som gäller för alla människor, även icke-medborgare
Skidåkning	RYF-QPR-HGB-AYA	3	-
Volleyboll	RYF-QPR-HGB-BAH	3	-
Ishockey	RYF-QPR-HGB-BQN	3	-
Golf	RYF-QPR-HGB-CIS	3	-
Simning	RYF-QPR-HGB-CYD	3	-
Fotboll	RYF-QPR-HGB-DEN	3	-
Bowling	RYF-QPR-HGB-DGM	3	-
Bilsport	RYF-QPR-HGB-DNB	3	-
Amerikansk fotboll	RYF-QPR-HGB-DZQ	3	-
E-Sport	RYF-QPR-HGB-EDX	3	-
Motorcykelsport	RYF-QPR-HGB-EOW	3	-
Trav	RYF-QPR-HGB-FMU	3	-
Cykling	RYF-QPR-HGB-FUZ	3	-
Boxning	RYF-QPR-HGB-GKR	3	-
Orientering	RYF-QPR-HGB-GLR	3	-
Kanot	RYF-QPR-HGB-HNS	3	-
Ridsport	RYF-QPR-HGB-IPS	3	-
Bordtennis	RYF-QPR-HGB-JDC	3	-
Gymnastik	RYF-QPR-HGB-KJR	3	-
Padel	RYF-QPR-HGB-LHA	3	-
Basket	RYF-QPR-HGB-MII	3	-
Skridskoåkning	RYF-QPR-HGB-NYM	3	-
Bandy	RYF-QPR-HGB-OQU	3	-
Handboll	RYF-QPR-HGB-PJR	3	-
Friidrott	RYF-QPR-HGB-PWE	3	-
Innebandy	RYF-QPR-HGB-QHJ	3	-
Brottning	RYF-QPR-HGB-RQL	3	-
Tennis	RYF-QPR-HGB-WVD	3	-
Sportskytte	RYF-QPR-HGB-YDK	3	-
Kampsport	RYF-QPR-HGB-YGB	3	-
Skidskytte	RYF-QPR-HGB-YKG	3	-
Futsal	RYF-QPR-HGB-YOC	3	-
Badminton	RYF-QPR-HGB-YTM	3	-
Heminredning	RYF-TKT-AGB-BGD	3	-
Renovering	RYF-TKT-AGB-NVR	3	-

Trädgård	RYF-TKT-AGB-OKN	3	-
Dryck	RYF-TKT-ESN-JPB	3	-
Mat	RYF-TKT-ESN-QBA	3	-
Bakning	RYF-TKT-ESN-XAE	3	-
Smink	RYF-TKT-GSR-BIH	3	-
Hårmode	RYF-TKT-GSR-CNN	3	-
Kläder & skor	RYF-TKT-GSR-EXJ	3	-
Bilentusiasm	RYF-TKT-MRA-KWO	3	-
MC-entusiasm	RYF-TKT-MRA-TJY	3	-
Datorspel	RYF-TKT-VVD-EME	3	En form av interaktiv underhållning som spelas med hjälp av till exempel en persondator, spelkonsol eller mobiltelefon
Styrketräning	RYF-TKT-YIG-PME	3	-
Konditionsträning	RYF-TKT-YIG-RYB	3	-
Avfall	RYF-VHD-IFE-EGU	3	-
Farliga ämnen & strålning	RYF-VHD-IFE-FHW	3	Ämnen skadliga för människor och djur. Till exempel giftgaser, strålning, kemikalier, tungmetaller och PCB
Miljösanering	RYF-VHD-IFE-FKG	3	-
Naturföroreningar	RYF-VHD-IFE-LRF	3	-
Ekosystem	RYF-VHD-OAY-CBT	3	Ett system av växter, djur och bakterier relaterade till varandra i sin fysikaliska eller kemiska miljö
Djur	RYF-VHD-OAY-DLT	3	-
Hotade arter	RYF-VHD-OAY-VCS	3	Arter som riskerar att försvinna, till stor del på grund av förändringar i miljön, jakt, eller väder
Hav	RYF-VHD-QTT-RVR	3	-
Älvar & åar	RYF-VHD-QTT-VWZ	3	-
Miljöfrämjande arbete	RYF-VHD-ZKU-HLP	3	-
Energibesparing	RYF-VHD-ZKU-SBV	3	-
Smittsam sjukdom	RYF-WUU-JZK-FBC	3	-
Funktionshinder	RYF-WUU-JZK-TAC	3	-
Missbruksvård	RYF-WUU-JZK-WLI	3	-
Cancer	RYF-WUU-JZK-YIA	3	-
Hjärt- & kärlsjukdomar	RYF-WUU-JZK-YMA	3	-
Psykisk ohälsa	RYF-WUU-JZK-YSP	3	-
Mediciner	RYF-WUU-QNB-LKJ	3	-
Medicinska tester	RYF-WUU-QNB-OFX	3	-
Vacciner	RYF-WUU-QNB-RCV	3	-
Specialistvård	RYF-WUU-SUX-APJ	3	-
Primärvård	RYF-WUU-SUX-CJL	3	-
Tandvård	RYF-WUU-SUX-CQS	3	-
Äldreomsorg (Sjukvård)	RYF-WUU-SUX-MEF	3	-
Omsorg	RYF-WUU-SUX-SFB	3	-
Demonstrationer	RYF-WXT-BND-ATO	3	En demonstration är en folkmassa som samlats för att offentligt uttrycka en opinion i en politisk eller annan fråga
Tv-serier	RYF-XKI-BUS-IFX	3	Till exempel "Tre Kronor", "Bonusfamiljen"
Reality-tv	RYF-XKI-BUS-NXX	3	Till exempel "Big Brother", "Gift vid första ögonkastet"
Underhållningsprogram	RYF-XKI-BUS-NYP	3	Till exempel "På Spåret", "Quizza med P3"
Radio	RYF-XKI-BUS-RWA	3	Program som sänds av radiostationer
Sociala medier	RYF-XKI-BUS-RXO	3	-
Tidningar	RYF-XKI-BUS-SBN	3	-
Journalistik	RYF-XKI-BUS-TVB	3	-
Musikgenre	RYF-XKI-FEY-VVW	3	-

Revy	RYF-XKI-GJH-DLL	3	Dramaföreställningar som är en satir över något som hänt under det gångna året
Teater	RYF-XKI-GJH-LNA	3	-
Fotografi	RYF-XKI-LNE-FAE	3	-
Konsthantverk	RYF-XKI-LNE-QWZ	3	-
Skulptur	RYF-XKI-LNE-SNR	3	Konstruktioner i keramik, sten, trä, metall eller andra fasta material i konstsyfte
Måleri	RYF-XKI-LNE-UZQ	3	-
Teckning	RYF-XKI-LNE-VPL	3	-
Poesi	RYF-XKI-YFJ-HKG	3	-
Skönlitteratur	RYF-XKI-YFJ-XZQ	3	Litteratur som inte nödvändigtvis behöver baseras på fakta. Till exempel "Harry Potter" och "Arn"
Facklitteratur	RYF-XKI-YFJ-ZBW	3	Litteratur med fakta för att lära sig mer om ett ämne. Till exempel matematikböcker
Förskola	RYF-YDR-MHH-DJZ	3	Verksamhet med barnomsorg och utbildning för barn åren innan den egentliga skolgången
Grundskola	RYF-YDR-MHH-IRE	3	Skolformen där elever lär sig grundläggande kunskaper, bland annat läsande och räknelära
Gymnasieskola	RYF-YDR-MHH-LKU	3	En avgiftsfri och frivillig sekundärutbildning i Sverige för ungdomar som har gått ut grundskolan
Vidareutbildning	RYF-YDR-MHH-UAD	3	-
Betyg	RYF-YDR-WXH-EDS	3	-
Löner & förmåner	RYF-ZEI-CLV-AGH	3	-
Konflikter	RYF-ZEI-CLV-PSL	3	Skillnader i uppfattning om till exempel arbetsförhållanden eller löner
Kollektivavtal	RYF-ZEI-CLV-QGD	3	Vanligtvis skriftliga avtal som täcker en specifik klass av arbetare och deras anställnignsvillkor
Uppsägningar	RYF-ZEI-QWT-IIW	3	-
Arbetsmiljöfrågor	RYF-ZEI-RYD-YZX	3	Regler och rutiner för att garantera arbetstagarnas hälsa
Drograttfylleri	RYF-BIZ-WZJ-ELD-ASC	4	-
Olovlig körning	RYF-BIZ-WZJ-ELD-GUK	4	-
Vårdslöshet i trafik	RYF-BIZ-WZJ-ELD-JGD	4	-
Rattfylleri	RYF-BIZ-WZJ-ELD-QEG	4	-
Stöld	RYF-BIZ-WZJ-GNO-BUL	4	-
Inbrott	RYF-BIZ-WZJ-GNO-EBC	4	-
Tillgrepp av fortskaffningsmedel	RYF-BIZ-WZJ-GNO-HHO	4	-
Rån	RYF-BIZ-WZJ-GNO-JXB	4	-
Ringa stöld	RYF-BIZ-WZJ-GNO-UHT	4	-
Mordbrand	RYF-BIZ-WZJ-IQH-CUG	4	-
Allmänfarlig vårdslöshet	RYF-BIZ-WZJ-IQH-THQ	4	-
Allmänfarlig ödeläggelse	RYF-BIZ-WZJ-IQH-TXV	4	-
Misshandel	RYF-BIZ-WZJ-KVS-AJP	4	-
Mord	RYF-BIZ-WZJ-KVS-KTU	4	-
Våld i nära relationer	RYF-BIZ-WZJ-KVS-NTL	4	-
Jaktbrott	RYF-BIZ-WZJ-OFE-MXU	4	-
Bedrägeri	RYF-BIZ-WZJ-RPI-UBX	4	-
Skattebrott	RYF-BIZ-WZJ-RPI-XPC	4	-
Olaga hot	RYF-BIZ-WZJ-SVK-EPV	4	-
Hot mot tjänsteman	RYF-BIZ-WZJ-SVK-XSF	4	-
Ofredande	RYF-BIZ-WZJ-SVK-YBM	4	-

Utpressning	RYF-BIZ-WZJ-SVK-ZJY	4	-
Sexuellt ofredande	RYF-BIZ-WZJ-XEY-EXA	4	-
Våldtäkt	RYF-BIZ-WZJ-XEY-FOH	4	-
Sexualbrott mot barn	RYF-BIZ-WZJ-XEY-MWO	4	-
Böter	RYF-BIZ-XSV-BII-HDU	4	-
Fängelse	RYF-BIZ-XSV-BII-KLB	4	-
Åtal	RYF-BIZ-XSV-JIV-DXW	4	-
Rättegång	RYF-BIZ-XSV-JTA-YUA	4	-
Överklagande av dom	RYF-BIZ-XSV-JTA-ZUJ	4	När dom inte accepteras av den som dömts, av åklagaren eller brottsoffer
Tåg	RYF-HPT-YGE-GUI-AZI	4	-
Väg	RYF-HPT-YGE-GUI-EHJ	4	-
Flyg	RYF-HPT-YGE-GUI-JGK	4	-
Sjöfart	RYF-HPT-YGE-GUI-YMN	4	-
Detaljplan	RYF-HPT-ZSS-WTL-JLR	4	-
Kretskort & datorkomponenter	RYF-IXA-KEV-BEL-MZV	4	-
Telekommunikationsutrustning	RYF-IXA-KEV-BEL-OCZ	4	-
Telekommunikationstjänster & trådlös teknik	RYF-IXA-KEV-BEL-TMN	4	-
Renovering & restaurering	RYF-IXA-KEV-OTN-RBK	4	Återställa egenskaper eller strukturer till ett tidigare bättre tillstånd
Husbyggnad	RYF-IXA-KEV-OTN-TSA	4	-
Stadsbyggnad	RYF-IXA-KEV-OTN-WNQ	4	-
Infrastrukturbyggnad	RYF-IXA-KEV-OTN-ZWU	4	-
Kafé	RYF-IXA-KEV-RFJ-ABR	4	-
Bar	RYF-IXA-KEV-RFJ-IDT	4	Ett företag där drycker tillagas och serveras till allmänheten för konsumtion på plats
Restaurang	RYF-IXA-KEV-RFJ-KXI	4	Ett företag där måltider tillagas och serveras till allmänheten
Elektronisk handel	RYF-IXA-KEV-RQL-AOK	4	-
Detaljhandel	RYF-IXA-KEV-RQL-IDW	4	-
Specialbutiker	RYF-IXA-KEV-RQL-LYC	4	-
Livsmedel	RYF-IXA-KEV-RQL-MGE	4	-
Internethandel & postorder	RYF-IXA-KEV-RQL-WBC	4	Objekt som beställs och levereras per post
Leksakshandel	RYF-IXA-KEV-RQL-WDB	4	-
Kläder	RYF-IXA-KEV-RQL-XNY	4	-
Solenergi	RYF-IXA-KEV-UBU-FLQ	4	-
Bensin	RYF-IXA-KEV-UBU-FVR	4	Nyheter om till exempel höjning av bensinpriser eller nedläggning av bensinmackar
Biobränslen	RYF-IXA-KEV-UBU-KHX	4	Energi som inte framställs av fossila bränslen
Vindkraft	RYF-IXA-KEV-UBU-KPP	4	-
Olja & gas	RYF-IXA-KEV-UBU-SOG	4	Nyheter om till exempel gasprospekt, oljehamnar och oljeborrning
Kärnkraft	RYF-IXA-KEV-UBU-XCR	4	-
Vattenkraft	RYF-IXA-KEV-UBU-YEW	4	-
Dieselbränsle	RYF-IXA-KEV-UBU-ZZD	4	-
Marknadsundersökningar	RYF-IXA-KEV-VKR-DTQ	4	En tjänst som försöker bestämma vad folk vill köpa
Bud- & transporttjänster	RYF-IXA-KEV-VKR-KGD	4	-
Auktionstjänster	RYF-IXA-KEV-VKR-KHT	4	Försäljning per budgivning
Banktjänster	RYF-IXA-KEV-VKR-LRI	4	Tjänster för lagring, överföring, mottagning och leverans av pengar
Bokföring & revision	RYF-IXA-KEV-VKR-MFC	4	Tjänster som erbjuder balansräkning av budget samt riktigheten av finansiella rapporter
Bemanningsföretag	RYF-IXA-KEV-VKR-MVU	4	En tjänst som hjälper människor att hitta jobb och företag att hitta arbetare

Försäkringar	RYF-IXA-KEV-VKR-ODH	4	-
Uthyrning	RYF-IXA-KEV-VKR-OHX	4	-
Servicetjänster	RYF-IXA-KEV-VKR-SKJ	4	Konsument tjänst som är immateriell, till exempel skönhetsvård eller frisör
Livsmedelsindustrin	RYF-IXA-KEV-VRA-BLM	4	-
Tobak	RYF-IXA-KEV-VRA-BOL	4	Odling, produktion och försäljning av tobaksvaror
Destillering & bryggeriverksamhet för alkoholhaltiga drycker	RYF-IXA-KEV-VRA-JBZ	4	Tillverkning av alkoholdrycker
Möbel- & hemindredningstillverkning	RYF-IXA-KEV-VRA-XAL	4	Tillverkning av möbler, tapeter, färger och tyger för inredning
Djuruppfödning	RYF-IXA-KEV-VSA-CJI	4	-
Jordbruk	RYF-IXA-KEV-VSA-GTD	4	-
Skogsbruk	RYF-IXA-KEV-VSA-MEC	4	-
Fiskerinäring	RYF-IXA-KEV-VSA-NQV	4	-
Järnvägstransporter	RYF-IXA-KEV-VXX-SAU	4	Verksamheten för att transportera personer eller gods på järnväg
Sjötransporter	RYF-IXA-KEV-VXX-TBB	4	Kommersiella transporter av människor eller gods via båtar, fartyg och vatten
Flygtransporter	RYF-IXA-KEV-VXX-TWI	4	Flygplan och flygplatsverksamhet
Vägtransporter	RYF-IXA-KEV-VXX-YRQ	4	Verksamheten att transportera gods med lastbil och motorvägar
Tung industri	RYF-IXA-KEV-WHS-PFN	4	Tillverkare av kranar, bulldozers och liknande för användning i större byggprojekt
Fordonstillverkning	RYF-IXA-KEV-WHS-ZIB	4	-
Läkemedelsproduktion	RYF-IXA-KEV-XZV-PYI	4	-
Hälsa & skönhetsprodukter	RYF-IXA-KEV-XZV-QRS	4	-
Hotell & logi	RYF-IXA-KEV-YII-NFS	4	Verksamhet i form av mat och husrum för resenärer
Aktieutdelning	RYF-IXA-LHV-QOI-FMD	4	Meddelanden om aktieägarnas faktiska utdelning, använd också för vinstvarning
Konkurser	RYF-IXA-LHV-QOI-TDN	4	Nyheter om faktiska konkurs anmälningar
Bostadspriser	RYF-IXA-QED-UGD-FRC	4	-
Demokrati	RYF-KNI-OHT-PIO-PCQ	4	En regering som väljs av folket med mandat att utöva folkets vilja inom ramarna av mänskliga rättigheter
Diktatur	RYF-KNI-OHT-PIO-PNL	4	En regering som inte valts av folket och är fria att bedriva politik som främjar regeringens intressen
Samhällstjänst & allmännytta	RYF-KNI-RLW-HER-VNR	4	Obetald tjänst för samhället som civila
Säkerhetsåtgärder (försvar)	RYF-KNI-RLW-YIU-ADH	4	-
Väpnade styrkor	RYF-KNI-RLW-YIU-OVY	4	-
Offentliga upphandlingar	RYF-KNI-XNS-SWE-OTW	4	-
Integrationspolitik	RYF-KNI-XNS-SWE-OWE	4	-
Vapenpolitik	RYF-KNI-XNS-SWE-PEU	4	-
Bostads- & stadsplaneringsfrågor	RYF-KNI-XNS-SWE-ROA	4	-
Pension & välfärdspolitik	RYF-KNI-XNS-SWE-VMT	4	-
Budget	RYF-KNI-XNS-WZY-ZFI	4	Styrdokument för utgifter och investeringar ett styre skall göra under en period och hur de skall finansieras av lån eller skatt
Alpint	RYF-QPR-HGB-AYA-FLN	4	-
Längdåkning	RYF-QPR-HGB-AYA-MFT	4	-
Elitserien dam	RYF-QPR-HGB-BAH-KFA	4	-

SHL	RYF-QPR-HGB-BQN-BXE	4	-
Hockeytvåan	RYF-QPR-HGB-BQN-KNN	4	-
Hockeyallsvenskan	RYF-QPR-HGB-BQN-MFF	4	-
Hockeyettan Västra vår	RYF-QPR-HGB-BQN-PYW	4	-
J20	RYF-QPR-HGB-BQN-QPQ	4	-
Hockeytrean	RYF-QPR-HGB-BQN-TKF	4	-
Hockeyettan	RYF-QPR-HGB-BQN-TQL	4	-
SDHL	RYF-QPR-HGB-BQN-VFB	4	-
J18	RYF-QPR-HGB-BQN-VLP	4	-
NHL	RYF-QPR-HGB-BQN-ZBN	4	-
Division 4 herr	RYF-QPR-HGB-DEN-ALX	4	-
Division 2 dam	RYF-QPR-HGB-DEN-BMK	4	-
Svenska cupen herr	RYF-QPR-HGB-DEN-BUA	4	-
Division 6 herr	RYF-QPR-HGB-DEN-CWD	4	-
Division 3 dam	RYF-QPR-HGB-DEN-FXI	4	-
Allsvenskan	RYF-QPR-HGB-DEN-GAS	4	-
Division 1 herr	RYF-QPR-HGB-DEN-GZQ	4	-
Division 3 herr	RYF-QPR-HGB-DEN-HKY	4	-
Division 7 herr	RYF-QPR-HGB-DEN-HPJ	4	-
Elitettan	RYF-QPR-HGB-DEN-LMC	4	-
Division 5 herr	RYF-QPR-HGB-DEN-MVN	4	-
Division 4 dam	RYF-QPR-HGB-DEN-SZC	4	-
Division 2 herr	RYF-QPR-HGB-DEN-TQG	4	-
Division 1 dam	RYF-QPR-HGB-DEN-UBA	4	-
Superettan	RYF-QPR-HGB-DEN-WAM	4	-
Damallsvenskan	RYF-QPR-HGB-DEN-XNC	4	-
Rally	RYF-QPR-HGB-DNB-FHE	4	-
Indy Racing	RYF-QPR-HGB-DNB-LOH	4	-
Speedway	RYF-QPR-HGB-EOW-CBQ	4	-
Enduro	RYF-QPR-HGB-EOW-PSO	4	-
MTB	RYF-QPR-HGB-FUZ-DFH	4	-
Svenska Basketligan herr	RYF-QPR-HGB-MII-SVM	4	-
Division 1	RYF-QPR-HGB-OQU-BQZ	4	-
Svenska Cupen	RYF-QPR-HGB-OQU-GBI	4	-
Allsvenskan herr	RYF-QPR-HGB-OQU-HEN	4	-
Elitserien herr	RYF-QPR-HGB-OQU-KCB	4	-
Elitserien dam	RYF-QPR-HGB-OQU-LWB	4	-
Allsvenskan herr	RYF-QPR-HGB-PJR-AAK	4	-
Handbollsligan herr	RYF-QPR-HGB-PJR-GVY	4	-
SHE	RYF-QPR-HGB-PJR-RAE	4	-
Division 3 dam	RYF-QPR-HGB-PJR-ZDD	4	-
Långdistanslöpning	RYF-QPR-HGB-PWE-KRH	4	-
Allsvenskan herr	RYF-QPR-HGB-QHJ-CUA	4	-
Allsvenskan dam	RYF-QPR-HGB-QHJ-HBV	4	-
Division 2 herr	RYF-QPR-HGB-QHJ-OTJ	4	-
Superligan dam	RYF-QPR-HGB-QHJ-RVU	4	-
Division 2 dam	RYF-QPR-HGB-QHJ-SSA	4	-
Division 1 dam	RYF-QPR-HGB-QHJ-UVW	4	-
Division 3 herr	RYF-QPR-HGB-QHJ-XFL	4	-
Superligan herr	RYF-QPR-HGB-QHJ-XQM	4	-
Division 1 herr	RYF-QPR-HGB-QHJ-XSZ	4	-
Futsalligan	RYF-QPR-HGB-YOC-TPJ	4	-
Rovdjur	RYF-VHD-OAY-DLT-XUZ	4	-
Epidemi	RYF-WUU-JZK-FBC-BSI	4	Ett utbrott av något, vanligen sjukdomar, som sprider sig mellan människor
Virussjukdomar	RYF-WUU-JZK-FBC-FRI	4	-
Pandemi	RYF-WUU-JZK-FBC-ORU	4	Epidemi som får spridning över stora delar av världen och drabbar många människor
Akutsjukvård	RYF-WUU-SUX-APJ-DZQ	4	-
Kirurgi	RYF-WUU-SUX-APJ-FCV	4	-
Psykiatri	RYF-WUU-SUX-APJ-GAB	4	-
Pediatrik (barnsjukvård)	RYF-WUU-SUX-APJ-RJV	4	-
Dansband	RYF-XKI-FEY-VVW-BJY	4	-

Folkmusik	RYF-XKI-FEY-VVW-CRI	4	-
Country & blues	RYF-XKI-FEY-VVW-ISA	4	Traditionella musikgenrer med rötter från den amerikanska södern
Pop	RYF-XKI-FEY-VVW-NQX	4	-
Klassisk musik	RYF-XKI-FEY-VVW-OFH	4	Musik som följer klassiska strukturer rytm och harmoni
Jazz	RYF-XKI-FEY-VVW-QSI	4	-
Hiphop & RNB	RYF-XKI-FEY-VVW-XGJ	4	-
Rock	RYF-XKI-FEY-VVW-ZJU	4	-
Textil	RYF-XKI-LNE-QWZ-VTV	4	-
Högskola	RYF-YDR-MHH-UAD-FEL	4	-
Folkhögskola	RYF-YDR-MHH-UAD-JTM	4	-
Yrkesutbildning	RYF-YDR-MHH-UAD-LVQ	4	-
Universitet	RYF-YDR-MHH-UAD-VTV	4	Universitet är ett högre lärosäte med forskning och utbildning. Till exempel Mittuniversitetet och Uppsala universitet
Strejk	RYF-ZEI-CLV-PSL-DLO	4	Arbetstagares yttersta stridsåtgärd vid arbetskonflikter
Avtalsfrågor (löner)	RYF-ZEI-CLV-QGD-AMA	4	-
Avtalsfrågor (arbetsregler)	RYF-ZEI-CLV-QGD-FZM	4	-

Table C.2: List of all categories