Using Hierarchical Class Structure to Improve Fine-Grained Claim Classification

Erenay Dayanik[1], **André Blessing**[1], **Nico Blokker**[2], **Sebastian Haunss**[2], **Jonas Kuhn**[1], **Gabriella Lapesa**[1], and **Sebastian Padó**[1]

[1]IMS, University of Stuttgart, Germany
[2]SOCIUM, University of Bremen, Germany

Abstract

The analysis of public debates crucially requires the classification of political demands according to hierarchical *claim ontologies* (e.g. for immigration, a supercategory "Controlling Migration" might have subcategories "Asylum limit" or "Border installations"). A major challenge for automatic claim classification is the large number and low frequency of such subclasses. We address it by jointly predicting pairs of matching super- and subcategories. We operationalize this idea by (a) encoding soft constraints in the claim classifier and (b) imposing hard constraints via Integer Linear Programming. Our experiments with different claim classifiers on a German immigration newspaper corpus show consistent performance increases for joint prediction, in particular for infrequent categories and discuss the complementarity of the two approaches.

1 Introduction

Newspaper articles are an invaluable source for the analysis of public debates. In political science, it is common to manually annotate the articles by identifying *claims* (text spans which report a demand on a specific policy aspect), assigning them fine-grained *claim categories* from domain-specific *claim ontologies* and attributing them to *actors* (e.g., politicians or parties). Actors and claim categories together can be used to construct expressive *discourse networks* (Leifeld, 2016) for in-depth analysis of debate structure and dynamics.

In line with the trend of applying NLP methods to questions from political science (e.g., Bamman and Smith, 2015; Glavaš et al., 2019) claim classification has been framed as generic text classification (Padó et al., 2019). That study however addresses only coarse-grained categories and reports mixed results even at that level, with a macro F1 of 46. This is arguably due to the well-known problems of fine-grained classification: The larger the set of classes, the more data would be desirable, while in actuality, the number of instances per class shrinks (Mai et al., 2018; Chang et al., 2020).

Our paper aims at developing practically useful models of fine-grained claim classification. Its main proposal is to exploit the hierarchical nature of claim ontologies by jointly predicting (frequent) supercategories and (informative) subcategories. By enforcing consistency between the levels, the predictions can profit off each other. We experiment with two operationalizations of this idea. The first one, Hierarchical Label Encoding (HLE, Shimaoka et al. (2017a)) introduces "soft" constraints through parameter sharing between classes in the classifier. The second one, Integer Linear Programming (ILP, e.g., Punyakanok et al. (2004)) introduces "hard" constraints in a post-processing step.

Both methods can be applied to a range of claim classifier architectures. We present experiments with four architectures on a German manually annotated corpus from the so-called refugee crisis in Germany in 2015. We answer the following questions: Do HLE and ILP improve the performance in our experimental setup? (Yes.) Is there complementarity between them? (Yes.) Does the effect depend on the underlying architectures. (Broadly, no.) What types of classes is the improvement most pronounced for. (Low-frequency ones.)

2 Dataset and Claim Ontology

Our experiments are conducted on an extended version of the *DebateNet-migr15* (Lapesa et al., 2020).[1] This corpus comprises 1361 articles published in 2015 on the German quality newspaper *taz*. The corpus is annotated manually according to a two-level claim ontology developed by politi-

[1]For details on the availability of the dataset and code used in our experiments, see `mardy-spp.github.io`

Proceedings of the 5th Workshop on Structured Prediction for NLP, pages 53–60

Original corpus

Code	Label	f	n.sub	mean f.sub
1xx	Controlling Migration	998	16	62 ± 46.2
2xx	Residency	726	18	40 ± 41.2
3xx	Integration	475	15	31 ± 35.5
4xx	Domestic Security	230	9	25 ± 17.9
5xx	Foreign Policy	689	9	76 ± 17.8
6xx	Economy	194	12	16 ± 13.1
7xx	Society	749	19	39 ± 37.9
8xx	Procedures	676	20	33 ± 37.7

(a)

Modified corpus

Code	f	low	mid	high
1xx	994	2	3	7
2xx	726	4	4	4
3xx	470	1	3	2
4xx	229	3	3	0
5xx	686	3	2	3
6xx	192	3	2	0
7xx	744	4	3	5
8xx	667	5	3	3

(b)

Table 1: (a): Claim distribution by supercategories: *Code*; *Label*; frequency (*f*); number of subcategories (*n.sub*); mean subcategory frequency with SD (*mean f.sub*). (b): Claim distribution for each supercategory after very infrequent classes are merged. low/mid/high represents the distribution of subcategory frequencies.

cal science experts for the migration domain. The corpus contains 3827 annotated textual spans, each of which is assigned one or more categories from the claim ontology described below: spans can be assigned multiple categories when the statements touch on more than one policy issue.

Claim ontology Policy debates are inherently complex, as a reflection of the complexity of the problems which the policy addresses: in our case, control of migration, but also integration of refugees, foreign policy, etc. In our case, the claim ontology consists of 100 subcategories which are grouped into 8 supercategories (cf. Table 1a). For example, 'border controls' and 'quota for refugees' are subcategories of the supercategory 'migration control'. The fine-grained annotation is crucial to build a satisfactory picture of a policy debate: what we are interested in is the position of certain politicians with respect to specific policy aspects over time (i.e., being in favor or against refugee quotas), while the supercategories are not expressive enough for the analysis of the debate itself. At the same time, Table 1a shows the drop in frequency between supercategories (in the hundreds) and subcategories (in the tens), with pronounced differences between categories, resulting in a clear modeling challenge. We return to this point in Section 5.

3 Basic Claim Classification

Given the properties described above, we model claim classification as multi-label classification. We follow previous work on coarse-grained claim classification (Padó et al., 2019) in comparing a set of neural models, ranging from baselines to state-of-the-art architectures. All models are trained using cross entropy loss with the sigmoid activation function. All models except BERT use custom FastText (Bojanowski et al., 2017) word embeddings pretrained on a German newswire corpus.[2]

LSTM This model passes the input through a single-layer LSTM. The final hidden state is used as input to a fully connected layer.

BiLSTM A single-layer Bidirectional LSTM (Graves et al., 2013) traverses the input. The final hidden states in both directions are concatenated and fed to a fully connected layer.

BiLSTM+Attention This model combines the BiLSTM architecture with the attention mechanism described in Shimaoka et al. (2017a). The input is fed to a single-layer BiLSTM. Then, the attention-weighted sum of the hidden states corresponding to the input sequence is fed to a fully connected layer.

BERT This is a pretrained BERT (Devlin et al., 2019) model trained solely on German corpora [3] and a fully connected layer which is trained while the BERT encoder is fine-tuned. After each input is encoded, we use the final hidden state of the first token, corresponding to the special token [CLS], as the contextualized representation of the input which serves as input to a fully connected layer.

[2]Further details regarding the architecture and training parameters can be found in the appendix.

[3]https://deepset.ai/german-bert

4 Integrating Hierarchical Class Structure

The obvious shortcoming of the model architectures sketched above is that they make the standard assumption of class independence – even though we know that the classes in claim classification are related. We therefore build on the idea that we can label all documents with both sub- and supercategories during training time, and then encourage the model to jointly predict categories at both levels *so that these predictions are consistent with one another*. The expectation is that this creates an incentive to learn better representations for the fine-grained classes. We now sketch two generally applicable methods that implement this idea.

Hierarchical Label Encoding (HLE). The idea behind this approach is to inject the inference relation between sub- and supercategories into the representation learning process. Following Shimaoka et al. (2017a), we create a binary square matrix, $S \in \{0,1\}^{l \times l}$, where l is the number of claim classes in dataset. Each cell in the matrix is filled with 1 either if the column class is subclass of or same as the row class, and filled with 0 otherwise. The matrix S is not updated during training and integrated into models by multiplying it by the weight matrix W of the final fully connected layer of each model: $p(y = 1) = \text{sigm}(h(W^\top S)^\top)$ where $W \in \mathbb{R}^{l \times hs}$, $h \in \mathbb{R}^{1 \times hs}$, $|y| = l$, and hs is the size of the hidden state of (Bi)LSTM or BERT. HLE introduces parameter sharing between classes in the same hierarchy (e.g. 100 and 101), but does not guarantee that the prediction output contains both a super- and a subcategory.

Integer Linear Programming (ILP). ILP has been applied to enforce linguistically motivated constraints on predicted structures such as semantic roles (Punyakanok et al., 2004), dependency parsing (Riedel and Clarke, 2006), or entailment graphs (Berant et al., 2011). Formally, an integer linear program is an optimization problem over a set of integer variables $\mathbf{x}$, given a linear objective function with a set of coefficients $\mathbf{c}$ and a set of linear inequality (and equality) constraints (Schrijver, 1984):

$$\max \mathbf{c}^\intercal \mathbf{x} \qquad \text{so that } A\mathbf{x} \geq b$$

We use ILP to select the most likely legal output from the probabilities estimated by the classifiers. Legal outputs are those where (a) for each predicted subcategory, the matching supercategory is predicted, and (b) for each predicted supercategory, at least one matching subcategory is predicted. We introduce a binary variable x_i for each supercategory and subcategory in the claim ontology, indicating whether this class is being predicted. This makes our task a binary optimization problem, a subclass of ILP. The coefficients c are given by the probability estimates of the neural claim classifiers (NCCs):

$$c_i = P_{\text{NCC}}(x_i = 1)$$

The objective function is the log likelihood of the complete model output, including both predicted and non-predicted classes:

$$\sum_i \log c_i x_i + \log[1 - c_i](1 - x_i)$$

The first constraints we impose on the solution is that each predicted subcategory must be accompanied by the matching supercategory. Let sup(i) denote the supercategory for the subcategory i. Then this constraint can be formalized as:

$$\text{for each subcategory } x_i : x_i - x_{sup(i)} \leq 0$$

The second constraint is that each predicted supercategory is accompanied by at least one if its subcategories. Let subs(i) denote the set of subcategories for supercategory i. The constraint is:

$$\text{for each supercategory } x_i : x_i - \sum_{j \in subs(i)} x_j \leq 0$$

ILP has a complementary profile to HLE in enforcing hard constraints on the output, without propagating the errors back to representation learning.

5 Experimental Evaluation

Setup. We remove very infrequent subcategories in the dataset by applying a threshold of 20 instances. Smaller categories are merged with the preexisting subcategory x99, which exists for each supercategory as a 'catch-all' category for outlier cases. After filtering, there are 8 super- and 72 subcategories left in the dataset (cf. Table 1b). We experiment with four model variations: **Plain** (base claim classifiers as in Section 3); **ILP** and **HLE** as described in Section 4; and **ILP+HLE**.

We split our dataset to train (90%) test (10%) splits and run the experiments on our own cluster with two Nvidia GeForce 1080GTX Ti GPUs. For

each experiment, we perform grid search guided by cross-validation on the training set to find the best hyperparameters. We report Precision, Recall and F1 scores weighted over all subcategories.

Main Results. Table 2 summarizes the results of our experiments. In the 'plain' setting, LSTM and BiLSTM perform significantly worse than BiLSTM+Attention and BERT. This finding is consistent with the generally observed benefit of attention and previous results by Padó et al. (2019).

The addition of ILP (2nd column) leads to inconsistent changes in precision but always yields better Recall and F-Scores. LSTM and BiLSTM still perform significantly worse than the other two models. When we switch to HLE, all metrics for all models are boosted significantly, showing that parameter sharing via the super/sub-category co-occurrence matrix is a successful across the board. We observe the largest improvement for BERT, where HLE yields an improvement of 12 points in F1, and leads to the overall highest Precision (0.75).

The last column (HLE + ILP) shows a substantial complementarity of the two methods: models consistently improve over both the HLE only and ILP only setting. Specifically, HLE+ILP models achieves better Recall scores than HLE models (+7 points on average) and better Precision (+8 points on average) scores than ILP models. The effect is least pronounced for the best architecture (BERT); nevertheless, BERT with HLE and ILP achieves the overall highest Recall (0.59) and F-Score (0.60), corresponding to an improvement of 13 points F1 compared to the 'plain' version. The fact that the F1 boost is fueled mainly by Recall is particularly promising because optimizing for Recall is the best strategy when NLP tools are employed in semi-automatic annotation (Ganchev et al., 2007; Ambati et al., 2011).

Frequency Band Analysis. As discussed in the introductory section, fine-grained classification struggles in particular with infrequent classes. We therefore ask how hierarchical class structure affects performance in relation to frequency. To do so, we analyze the performance of the best architecture (BERT), splitting the fine-grained categories into three equal-sized frequency bands.[4]

The results in Table 3 show that the prediction quality of plain BERT differs significantly across frequency bands. It fails badly in the low freq band (F1=0.1) while doing a fair job in the mid and high bands (F1=0.42 and 0.57, respectively). Again, we see consistent improvements for both ILP and HLE, but the improvements are more substantial for HLE, in particular for the low-freq band (+27 point F1). Combining HLE and ILP further increases Recall, but reduces Precision somewhat.[5]

In sum, we observe that both ILP and HLE improve fine-grained classification. The parameter sharing introduced by HLE particularly helps the lowest-frequency categories and increases both Precision and Recall. ILP generally boosts Recall by enforcing that both super- and a subcategories need to be predicted. There appears to be a mid-frequency "sweet spot" where this is particularly effective: Less frequent, and the probability estimates are not reliable enough; more frequent, the Precision–Recall trade-off is not worth it.

Qualitative Considerations. Finally, we investigate which subcategories benefit most from HLE and ILP in our best model (BERT). Table 4 again shows complementarity between HLE and ILP, indicating that a better combination of the two methods could lead to further improvements. HLE+ILP overlaps largely with HLE, mirroring the larger impact of HLE. Analysis of these classes shows that they belong to the mid and low frequency bands.

However, not all low and mid frequency classes profit equally. To explain this, we note that the fine-grained classes in the migration ontology differ substantially with regard to *concreteness*: While the high-level category 'Foreign policy' (5xx) contains relatively concrete sub-categories ('Enforcing Dublin III regulations' or 'Expanding the list of safe countries of origin'), the supercategory 'Society' (7xx) mostly consists of less manifest policy measures ('Uphold Human Rights', 'Oppose Xenophobia'). With regard to that distinction, the highest-gain subcategories are of the concrete kind (cf. Table 1): 106 ('Border defence'), 303 ('Forced integration'), 801 ('Constitutional law'), 807 ('Reducing bureaucracy'), 405 ('Counterterrorism'). Conversely, we do not find any subcategories of the less concrete supercategory 700 ('Society').

[4]Thresholds: high-frequency ($265 \geq f \geq 67$), mid-frequency ($65 \geq f \geq 40$) and low-frequency ($20 \geq f \geq 39$). Complete lists of the categories in the frequency bands and detailed results of other models are available in Table 5 and Table 6 in the appendix.

[5]We confirmed the relationship between frequency and performance with a correlation analysis to rule out a binning artifact. See Table 7 in the appendix.

Model	plain			ILP			HLE			HLE + ILP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM	0.50	0.24	0.30	0.45	0.28	0.32	0.60	0.33	0.39	0.52	0.38	0.41
BiLSTM	0.51	0.26	0.32	0.51	0.33	0.38	0.57	0.30	0.36	0.63	0.42	0.48
BiLSTM+Att	0.67	0.39	0.46	0.63	0.42	0.48	0.69	0.41	0.48	0.66	0.46	0.51
BERT	0.61	0.42	0.47	0.56	0.50	0.50	**0.75**	0.52	0.59	0.66	**0.59**	**0.60**

Table 2: Test results (weighted averages for fine-grained claim classification) for four architectures and two methods to integrate class structure (integer linear programming, hierarchical label encoding). Best results bolded.

Freq band	plain			ILP			HLE			HLE + ILP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Low freq	0.10	0.10	0.10	0.18	0.14	0.15	0.58	0.31	**0.37**	0.48	0.31	0.35
Mid freq	0.58	0.36	0.42	0.65	0.47	0.50	0.77	0.55	0.62	0.71	0.63	**0.65**
High freq	0.73	0.51	0.57	0.60	0.58	0.58	0.78	0.56	0.62	0.67	0.63	**0.64**

Table 3: Detailed results for BERT architecture: break down by frequency bands of fine-grained classes (highest F1 score for each frequency band bolded).

Setting	Highest Improvement
ILP	204, 499, 507, 508, 803
HLE	106, 303, 314, 801, 807
ILP+HLE	106, 303, 405, 801, 807

Table 4: Subcategories that gain most in F1 score

6 Conclusion

This paper has identified automatic fine-grained claim classification as a crucial, but underaddressed, component of political discourse analysis. We have demonstrated that hierarchical class structure can be exploited to lift fine-grained claim classification to a usable level, showing robust improvements even for transformer architectures and in particular for low-frequency claim categories.

Addressing the low-frequency issue is particularly relevant in the broader context of the goals of political science. Political discourse unfolds over time, and every prominent issue starts out as infrequent. The true dynamics of debates can only be captured if the classifiers are able to pick up the less salient categories (Koopmans and Statham, 1999; Kossinets, 2006). Future work involves investigating these concerns on a wider range of datasets, as well as evaluating fine-grained claim classification for semi-automatic discourse network construction.

Acknowledgments

We acknowledge funding by Deutsche Forschungsgemeinschaft (DFG) through MARDY (Modeling Argumentation Dynamics) within SPP RATIO and by Bundesministerium für Bildung und Forschung (BMBF) through E-DELIB (Powering up e-deliberation: towards AI-supported moderation).

References

Bharat Ram Ambati, Rahul Agarwal, Mridul Gupta, Samar Husain, and Dipti Misra Sharma. 2011. Error detection for treebank validation. In *Proceedings of the 9th Workshop on Asian Language Resources*, pages 23–30, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

David Bamman and Noah A. Smith. 2015. Open extraction of fine-grained political statements. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 76–85, Lisbon, Portugal. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

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

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '20, page 3163–3171, New York, NY, USA. Association for Computing Machinery.

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

Kuzman Ganchev, Fernando Pereira, Mark Mandel, Steven Carroll, and Peter White. 2007. Semi-automated named entity annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 53–56, Prague, Czech Republic. Association for Computational Linguistics.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2019. Computational analysis of political texts: Bridging research efforts across communities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 18–23, Florence, Italy. Association for Computational Linguistics.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE.

Ruud Koopmans and Paul Statham. 1999. Political Claims Analysis: Integrating Protest Event and Political Discourse Approaches. *Mobilization: An International Quarterly*, 4(2):203–221.

Gueorgi Kossinets. 2006. Effects of missing data in social networks. *Social Networks*, 28(3):247–268.

Gabriella Lapesa, Andre Blessing, Nico Blokker, Erenay Dayanik, Sebastian Haunss, Jonas Kuhn, and Sebastian Padó. 2020. DEbateNet-mig15: Tracing the 2015 immigration debate in Germany over time. In *Proceedings of LREC*, pages 919–927, Online.

Philip Leifeld. 2016. *Policy Debates as Dynamic Networks: German Pension Politics and Privatization Discourse*. Campus Verlag, Frankfurt/New York.

Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Tuan Duc Nguyen, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. 2018. An empirical study on fine-grained named entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 711–722, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sebastian Padó, Andre Blessing, Nico Blokker, Erenay Dayanik, Sebastian Haunss, and Jonas Kuhn. 2019. Who sides with whom? towards computational construction of discourse networks for political debates. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2841–2847, Florence, Italy. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1346–1352, Geneva, Switzerland. COLING.

Sebastian Riedel and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 129–137, Sydney, Australia. Association for Computational Linguistics.

Alexander Schrijver. 1984. *Linear and Integer Programming*. John Wiley & Sons, New York.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017a. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017b. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Appendix

A Results Details: Results by Frequency Bands for all Architectures

Table 5 presents Precision, Recall and F1 scores of models broken down for low, mid and high frequency bands. We observe similar patterns with other three models: (1) Prediction quality of models in plain setting differ significantly across frequency bands and all three models perform significantly worse on low frequency band and (2) Extending models with HLE and ILP leads to significantly better F1 scores on all frequency bands.

B Results Details: Correlation Analyses

We calculate Spearman's correlation coefficient in order to investigate the relationship between amount of available data for each subcategory and

Model	Freq band	plain			ILP			HLE			HLE + ILP		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM	Low freq	0.19	0.07	0.10	0.23	0.08	0.12	0.32	0.11	0.16	0.33	0.18	0.21
	Mid freq	0.57	0.20	0.28	0.53	0.25	0.31	0.66	0.33	0.42	0.58	0.38	0.43
	High freq	0.53	0.29	0.35	0.47	0.34	0.37	0.64	0.37	0.42	0.54	0.42	0.45
BiLSTM	Low freq	0.24	0.10	0.13	0.35	0.16	0.20	0.15	0.05	0.07	0.27	0.13	0.16
	Mid freq	0.54	0.20	0.28	0.56	0.27	0.35	0.68	0.22	0.32	0.54	0.30	0.37
	High freq	0.56	0.32	0.37	0.52	0.39	0.43	0.61	0.38	0.44	0.57	0.44	0.47
BiLSTM Att	Low freq	0.10	0.07	0.07	0.23	0.13	0.14	0.17	0.10	0.10	0.24	0.11	0.13
	Mid freq	0.82	0.47	0.56	0.69	0.45	0.53	0.71	0.46	0.53	0.64	0.52	0.55
	High freq	0.74	0.42	0.50	0.69	0.47	0.53	0.79	0.45	0.54	0.75	0.51	0.57
BERT	Low freq	0.10	0.10	0.10	0.18	0.14	0.15	0.58	0.31	0.37	0.48	0.31	0.35
	Mid freq	0.58	0.36	0.42	0.65	0.47	0.50	0.77	0.55	0.62	0.71	0.63	0.65
	High freq	0.73	0.51	0.57	0.60	0.58	0.58	0.78	0.56	0.62	0.67	0.63	0.64

Table 5: Detail results for all architectures by frequency band

performance change of BERT model across settings further. For that, we measure the difference between BERT model's subcategory performances in plan and other settings as well as amount of data available for each subcategory. Table 6 shows which subcategory belongs to which frequency band and Table 7 shows Spearman's correlation coefficients. We observe high negative values almost always indicating that there is a strong negative correlation between the amount of data exist for a subcategory and amount of change in performance which means that infrequent classes gain most from ILP and HLE.

Frequency Band	Label					
LOW	111	199	201	209	213	214
	406	408	499	502	505	508
	602	603	605	701	706	707
	708	801	802	807	811	814
MID	106	107	109	204	211	212
	215	301	302	303	307	401
	402	405	503	509	601	699
	702	711	715	803	804	808
HIGH	101	102	104	105	108	110
	190	202	203	207	299	309
	399	501	504	507	703	705
	709	712	799	805	812	899

Table 6: Lists of the categories in the frequency bands

	PAIR	P	R	F
BERT	Plain - ILP	-0.20	-0.10	-0.20
	Plain - HLE	-0.24	-0.31	-0.29
	Plain - (ILP+HLE)	-0.28	-0.29	-0.32

Table 7: Spearman's correlation coefficient results between change in evaluation metrics and subcategory size for BERT model.

C Training Details

In the LSTM model, we set the number of hidden units to 500. We train 300-dimensional FastText word embeddings on a corpus consisting of German Newspapers and use them as the input to LSTM. We use Adam with learning rate of 0.003 as optimizer. Batch size and number of epochs are set to 16 and 25 respectively.

In the BiLSTM model, we the set number of units to 500 in each direction and batch size to 16. The same 300-dimensional word embeddings as in the LSTM are used. The model is trained with Adam optimizer and a learning rate of 0.003 for 25 epochs.

In the BiLSTM+Attn model, we used the attention mechanism variant described in Shimaoka et al. (2017b). We set number of units to 500 in each direction and batch size to 16. We use the same 300-dimensional word embeddings used in LSTM and BiLSTM models, and train model for 20 epochs using Adam optimizer with learning rate of 0.003.

For the BERT model, we use a cased BERT variant[6] that was trained specifically for German with default parameters for the number of attention heads, hidden layers, and the number of hidden units are 12, 12, and 768, respectively. During fine-tuning, we use the Adam optimizer with learning rates of 5e-5, β_1 = 0.9, β_2 = 0.999, and set the maximum sequence length to 200, batch size to 16 and norm of maximum gradient to 1.0 and trained for 20 epochs.

Table 8 and Table 9 show the number of parameters in each model and average time required to

[6]https://deepset.ai/german-bert

	Parameter Numbers	
	Plain	HLE
LSTM	4,731,080	4,731,500
BiLSTM	6,375,080	6,376,000
BiLSTM Att	6,475,180	6,476,100
BERT	109,142,864	109,143,552

Table 8: Number of parameters in each model.

	Runtime (in Minutes)
LSTM	1.5
BiLSTM	2.2
BiLSTM Att	4.5
BERT	32.0

Table 9: Average runtime required to train each model

train each model used in our experiments respectively.

Hyperparameter search details We perform grid search for hyperparameter optimization and use the hyperparameters leading highest average F1 score during 5-Fold cross validation. Following lower and upper bounds have been applied during search for each hyperparameter: learning Rate [1e-4, 5e-2], epoch:[5, 25], batch size:[16, 32].

D Details about Dataset

Figure 1 depicts the number of instances for each category.

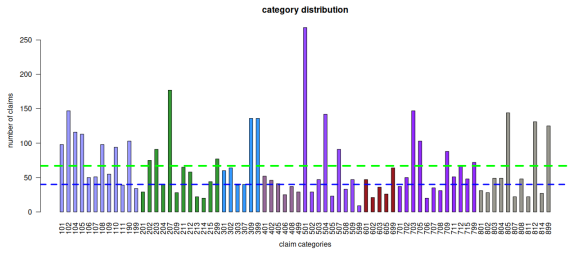

Figure 1: Claim distribution of subcategories. Green dotted line: boundary between high and mid frequency bands. Dark blue line: boundary between low and mid bands.