TraininG towards a society of data-saVvy inforMation prOfessionals to enable open leadership INnovation

MOVING

Using Deep Learning for Title-Based Semantic Subject Indexing to Reach Competitive Performance to Full-Texts

Florian Mai Lukas Galke Ansgar Scherp

Kiel University, Germany

Joint Conference on Digital Libraries, June 3-6, 2018, Fort Worth

www.moving-project.eu

Motivation: Titles vs. Full-Text

Good automatic semantic subject indexing methods based on metadata needed.

- Full-text not always available for text mining
- Metadata such as the title almost always available
- Title not competitive to full-text when the same number of training data is used [Galke et al., 2017]
- But far more labeled title samples (millions!) available than full-text data (several 100k)

Main Research Question

When all available titles are used, can deep learning close the performance gap between titles and full-texts?

- $> 650k$ samples: deep learning outperforms traditional methods at text classification [Zhang et al., 2015].
- Deep learning for text classification [Zhang et al., 2015, Yang et al., 2016, Grave et al., 2017, Liu et al., 2017]
- Multi-label text classification [Huang et al., 2011, Rubin et al., 2012, Nam et al., 2014, Große-Bölting et al., 2015, Galke et al., 2017]

- Employ a representative of each of the most common families of neural networks: MLPs, CNNs, LSTMs
- Frame subject indexing as multi-label classification problem
- All architectures share the same training procedure

Training Procedure

- Sigmoid at output layer: get output p_l for label l
- Minimize binary cross-entropy loss with Adam
- Assign label if $p_l > \theta$
- Tune θ on validation set during training
- Early stopping

Base-MLP [Galke et al., 2017] *(Baseline)*

- TF-IDF bag-of-words with 25,000 most frequent unigrams
- 1 hidden layer with 1,000 units
- dropout after hidden layer with rate 0.5

MLP

- additionally 25,000 most frequent bigrams
- wider layers and deeper networks
- Batch Normalization when beneficial

CNN and LSTM

CNN

- 1 layer of 1D-convolution [Kim, 2014] over the text with different window sizes (2, 3, 4, 5, 8)
- dynamic max-pooling [Liu et al., 2017]
- plus fully-connected layer

LSTM

- "vanilla" LSTM [Greff et al., 2017]
- bidirectionality and (self-)attention [Yang et al., 2016]

Sequence length is limited to 250 for LSTMs and CNNs.

A Note on Tuning the Methods

MOVING

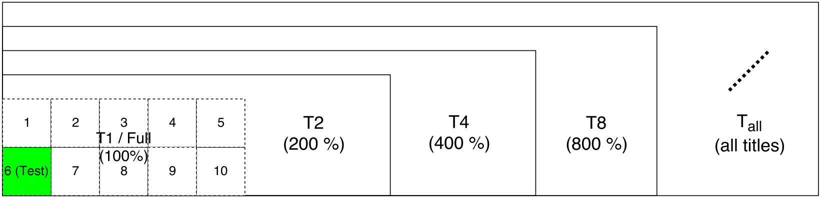
Remember!

We want to answer the question if the best title method can perform competitively to the best full-text method.

- Titles and full-texts are of different nature and therefore the best neural network architecture for titles is not necessarily the best for full-texts.
- For this reason, we tuned the architectures on one fold independently for titles and full-texts, resulting in very different solutions in some cases.

Experimental Setup

- $T1$, $T2$, $T4$, $T8$, T_{all} work with titles, *Full* works with full-texts.
- $T1$ and *Full* contain the same publications. Tx contains x times as many samples as $T1$.
- We split T1/Full into 10 folds and perform a 10-fold cross-validation

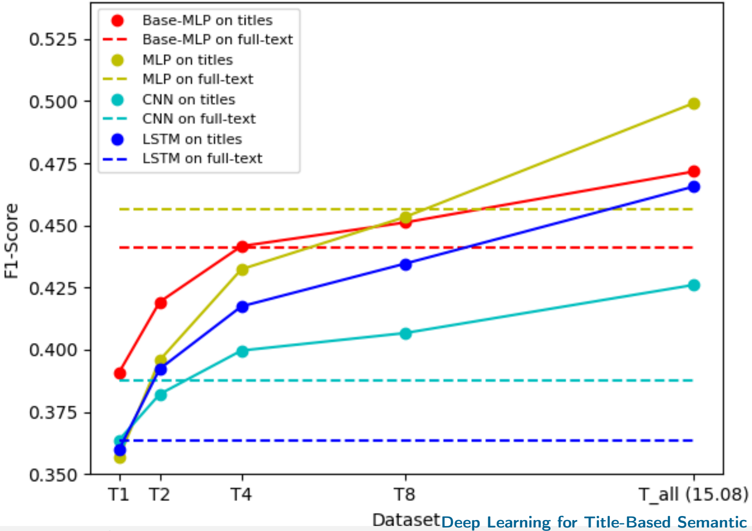


Datasets

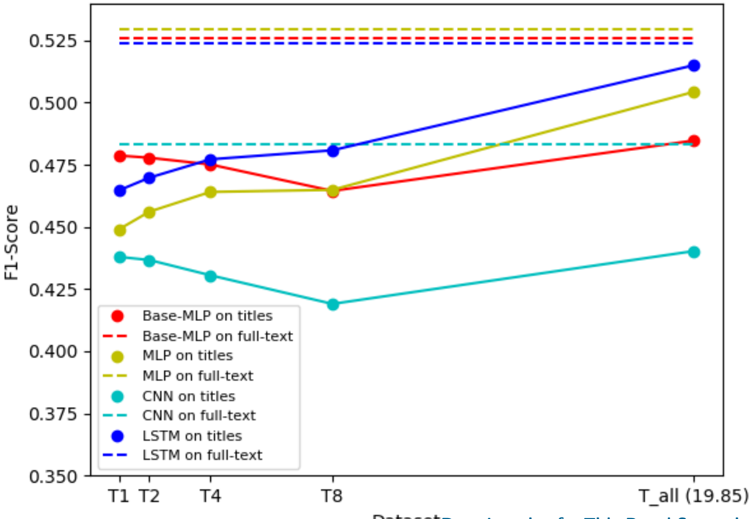
	EconBiz (STW)		PubMed (MeSH)	
	Title	Full-Text	Title	Full-Text
$\lvert D\rvert$	1,064,634	70,619	12,834,026	646,513

- Number of full-texts in PubMed $\approx 650k$!
- Number of full-texts in EconBiz $<< 650k$!

Results for EconBiz



Results for PubMed



Conclusion

Main Result

- Using all titles is at least competitive to using the full-text (titles 3% lower on PubMed and 9.4% higher on EconBiz).

Side Results

- The strategy to employ deep learning was largely successful since the more complex models tend to benefit more from additional samples.
- CNNs perform rather poor despite their prominence in text classification studies from recent years.

Thank You for Your Attention!

MOVING

Reproducibility

The source code, configurations, and title datasets can be found on GitHub: `https://github.com/florianmai/Quadflor`. Feel free to fork and run additional experiments!

Need more details?

An extended version of the paper (my master thesis) is also available online (see `https://github.com/florianmai/Quadflor`) and contains a lot more details:

- Intermediate results of tuning neural network architectures and hyperparameters

MOVING is funded by the EU Horizon 2020 Programme under the project number INSO-4-2015: 693092

References I

Galke, L., Mai, F., Schelten, A., Brunsch, D., and Scherp, A. (2017).
Using titles vs. full-text as source for automated semantic document annotation.
In *K-CAP*, page 9.

Grave, E., Mikolov, T., Joulin, A., and Bojanowski, P. (2017).
Bag of tricks for efficient text classification.
In *EACL*, pages 427–431.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017).
Lstm: A search space odyssey.
IEEE TNNLS.

References II

- Große-Bölting, G., Nishioka, C., and Scherp, A. (2015).
  A comparison of different strategies for automated semantic document annotation.
  In *K-CAP*, page 8. ACM.
- Huang, M., Névéol, A., and Lu, Z. (2011).
  Recommending mesh terms for annotating biomedical articles.
  JAMIA, 18(5):660–667.
- Kim, Y. (2014).
  Convolutional neural networks for sentence classification.
  In *EMNLP*, pages 1746–1751.
- Liu, J., Chang, W.-C., Wu, Y., and Yang, Y. (2017).
  Deep learning for extreme multi-label text classification.
  In *SIGIR*, pages 115–124. ACM.

References III

Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., and Fürnkranz, J. (2014).
Large-scale multi-label text classification - revisiting neural networks.
In *ECML*, pages 437–452.

Rubin, T. N., Chambers, A., Smyth, P., and Steyvers, M. (2012).
Statistical topic models for multi-label document classification.
JMLR, 88(1):157–208.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., and Hovy, E. H. (2016).
Hierarchical attention networks for document classification.
In *NAACL-HLT*, pages 1480–1489.

References IV

Zhang, X., Zhao, J., and LeCun, Y. (2015).
Character-level convolutional networks for text classification.
In *NIPS*, pages 649–657.

Choice of Hyperparameters

- MLP much more complex than Base-MLP: wider (2,000 units) or deeper (two layers with Batch Normalization)
- CNN uses large feature map size (400) except on one of the full-text datasets (100)
- Dynamic max-pooling only on full-texts, not beneficial on titles
- Multiple LSTM layers do not benefit the performance, but widening a single layer does (up to cell size 1,536)
- LSTMs are relatively small on full-texts (cell size 512 and 1,024, respectively), but larger on titles.

- Ideally: At each validation step, find the value for θ that yields the best F_1-score.
- However, doing a grid search over the entire space at each step is too costly.
- Observation: Increasing θ trades off recall for precision.
- We could make the assumption that for $\forall\theta : P(x) + R(x) = S$

F_1-score

$$F_1(x) = \frac{2 \cdot P(x) \cdot R(x)}{P(x) + R(x)} = \frac{2 \cdot P(x) \cdot R(x)}{S}$$

Tuning Threshold During Training II

- Under the assumption, $F_1(x)$ is a concave function of P and R with maximum at $P = R$.
- Heuristic approximation of best θ: At each validation step, we only consider a small neighborhood of the current value.
- Concretely: Start with $\theta_0 := 0.2$. At each step i,

$$\theta_i := \underset{\theta \in \{-k \cdot \alpha + \theta_{i-1}, \ldots, k \cdot \alpha + \theta_{i-1}\}}{\arg\max} F_1(D_{val}; c_i, \theta), \qquad (1)$$

where we set $\alpha = 0.01$ and $k = 3$, i.e., we try 7 threshold values at each step.

Explaining the Performance Drop

The distributions of T1/Full and T_{all} are quite different:

- T1/Full tend to come from more recent years than T_{all}.
- As a result, the label distributions differ, too.
- Training set of T2, . . . , T_{all} contain many labels that never appear in the test set.

	EconBiz			PubMed		
	|L|	rel. gain	abs. gain	|L|	rel. gain	abs. gain
T1	4,849	-	-	26,267	-	-
T2	5,165	6.5%	316	27,135	3.3%	868
T4	5,230	1.3%	65	27,447	1.1%	312
T8	5,357	2.4%	127	27,626	0.7%	179
T_{all}	5,661	5.7%	304	27,773	0.5%	147