Multi-Label Document Classification: Compensating For Information Scarcity in Titles With Larger Sample Sizes Master Thesis Proposal

Florian Mai

1 Introduction

Semantic annotations are crucial for users of digital libraries as they enhance the search of scientific documents. Given the large amount of new publications, automatic annotation algorithms are a useful tool for human expert annotators. However, due to legal barriers, often the full-text is unavailable for automatic processing. To address this issue, Galke et al. [7] have applied various methods that use only the title of a document and compared the results to those obtained from the same methods applied on full-texts. Even though they found that titles yield satisfactory results for most datasets, almost all methods were still substantially better when trained on full-texts. This is not a surprising result since overall a full-text contains more information and therefore also more indication of the document's topic. A human expert will always make better annotations based on the full-text.

However, we argue that for real world applications there is a chance the gap between titles and full-texts can be further narrowed or even closed. There are three main reasons for this conjecture.

First, much larger sample sizes can be leveraged for building a model. Mathematically, larger samples are guaranteed to improve the classification performance. For text classification specifically, the increase of the sample size may allow the training of powerful deep learning models, which are known to require very large amounts of data. Recently, Zhang et al. [16] developed a large convolutional neural network architecture for text classification. They evaluated on a variety of large datasets, ranging from 120,000 to 3,600,000 training samples and compared with a number of traditional models, including multinomial logistic regression based on bag-of-ngrams with TFIDF. On the four datasets with 560,000 training samples or less, the traditional approaches outperformed the proposed neural network architecture. On the remaining four datasets with 650,000 training samples or more, however, the neural network approach was superior.

Second, we can assume that full-texts contain a lot more passages of texts which are not directly indicative of the topic, even in scientific literature. While often authors choose the title such that it captures the key elements of the article, certain phrases in the full-text may appear independ of the topic. Bag-of-words models and CNNs, which are often considered great extractors of key-phrases indicative of a class, may be less prone to such phrases, as they learn to ignore them. Recurrent neural networks, on the other hand, may suffer more as they struggle to identify long-term dependencies within text. This is demonstrated and addressed in [5]. This problem could be even more severe if the text contains many phrases that are not indicative. We therefore argue that recurrent neural networks in particular benefit from the higher information density in titles as well as the much shorter length.

Lastly, due to the smaller information density in full-text, it is reasonable to expect that an estimator trained on the full-text needs to process a larger total amount of text to reach a fixed performance than an estimator trained on titles needs to reach that same performance. This could potentially impose high computational cost, so a model can not be trained in reasonable time. Especially recurrent neural networks can suffer because they have less op-

tions for parallelization. In that case, a smaller, less expressive model has to be chosen for faster convergence, which ultimately hurts the performance.

In order to study this question, we conduct experiments on three large-scale datasets $\mathcal{D} = \{D_1, D_2, D_3\}$ obtained from digital libraries. These datasets cover different domains, such as economics, medical science, and political science. We employ a diverse set of three classifiers $\mathcal{C} = \{c_1, c_2, c_3\}$ which have emerged from deep-learning research in recent years. In natural language processing, different types of neural networks have been successfully employed on different tasks. For text classification in particular, however, it is an open question whether convolutional neural networks, recurrent neural networks, or fully-connected feed-forward neural networks are superior. Therefore, we employ a representative of each type in our study.

Consider a dataset of document identifiers $D = \{d_1, \ldots, d_n\} \in \mathcal{D}$ and the corresponding titles $D_t = \{t_1, \ldots, t_n\}$ and available full-texts $D_f = \{f_1, \ldots, f_m\}$. Assuming $m < n$, we imply that we have less full-texts available than titles. Moreover, let $F_1^c(D')$ denote the classification performance of classifier $c \in \mathcal{C}$ in terms of the sample-based F_1 measure when using training corpus $D' \subseteq D$. We would like to answer the following research questions:

RQ1 For each dataset D, is there a factor x small enough such that a subset of the titles $D'_t \subseteq D_t$ with $|D'_t| = |D_f| \cdot x$ satisfies $\max_{c \in \mathcal{C}} F_1^c(D'_t) = \max_{c' \in \mathcal{C}} F_1^{c'}(D_f)$?

Intuitively speaking, can we obtain the best full-text classification performance with titles by increasing the training sample size?

RQ2 How fast does the classification performance increase with the number of titles? Formally, for fixed $D \in \mathcal{D}$, how does the ratio $r_i^{(1)} = \frac{\max_{c \in \mathcal{C}} F_1^c(D_{t_i})}{\max_{c' \in \mathcal{C}} F_1^{c'}(D_f)}$ develop with increasing $x_1 < \ldots < x_k$ where $|D_{t_i}| = |D_f| \cdot x_i$?

RQ3 What are the differences between the classifiers with respect to their performance and size of the training sample size? Formally, we assess the development of the ratio $r_i^{(2)} = \frac{F_1^c(D_{t_i})}{\max_{c' \in \mathcal{C}} F_1^{c'}(D_f)}$ with increasing $x_1 < \ldots < x_k$ for all D and classifiers $c \in \mathcal{C}$.

In summary, we conduct a study of *the relative classification performance on titles compared to full-text* as the dependent variable. We evaluate its behavior with respect to the independent variables *training sample size of titles*, the *dataset*, and the *classifier*.

2 Methodology

We discuss the methods and datasets we will explore in our research, as well as the experimental procedure to address the research questions.

Table 2 shows a timeline for the project. First, we build the datasets, so the interim evaluation of methods during their development is not biased towards one dataset. The following three and a half months will be occupied by incrementally improving on the basic models of each neural network type. Instead of sequentially developing one model after another, we will develop a new version of each of the subsequently described models every month. Lastly, we run and evaluate the final experiments in the remaining two months of the project. In Table 2, we present the mandatory and optional tasks to be tackled during the research period.

2.1 Classifiers

In the introduction, we argued that neural networks have a great potential to benefit from a vastly increasing sample size. Therefore, we employ a number of neural methods motivated from recent literature. Specifically, we consider a representative of each multi-layer-perceptrons, convolutional neural networks, and recurrent neural networks.

We start with a basic architecture for each type, which we consider mandatory for our proposed research. We then extend the basic version incrementally with techniques reported in the literature and identify the best configuration through experimentation. In the end, we only report results of the best

Month	June	July	August	September	October	November
Activity	Build datasets	Method Development			Final Experiments & Evaluation	
	Write Master Thesis					

Table 1: Timeline for the six months of the master thesis project.

	Mandatory	Optional
MLP	Base-MLP [7], word n-grams, lookup-table [8]	Morphological information [2]
CNN	Word-based single-layer CNN [10]	Multi-layer character-CNN or word-CNN [16, 4], Dynamic k-Max Pooling[9]
RNN	Word-based single-layer LSTM [16]	Dropout [6, 15], attention [1], HN-ATT [13]
Multi-label	Fixed threshold [7]	Predict labels with RNN [3]
Datasets	Economics, PubMed / NYT	Political science
Evaluation	Calculation of ratios and visual analysis	Regression analysis of functional form

Table 2: Tasks during the research period listed by obligation.

configuration to avoid a combinatorial explosion in the number of experiments. However, if we find that different architectures outperform each other depending on the type of text (title or full-text), we will report both to avoid a bias towards one type of text.

2.1.1 Multi-Layer-Perceptron

As a representative of the family of fully-connected neural networks, we employ a *multi-layer-perceptron (Base-MLP)* inspired by Nam et al. [11]. In their comparison of classifiers, Galke et al. [7] found this architecture to outperform all other (non-neural) classifiers on 7 out of 8 datasets. All the presented classifiers are based on the bag-of-words (BoW) feature representation, a traditionally strong baseline for text classification that disregards word order. Due to clearly superior performance, Base-MLP can also be considered the best representative of traditional BoW models. We therefore report its performance as our baseline for all subsequently developed models.

We will also aim to develop a more sophisticated version (*Soph-MLP*) of Base-MLP by incorporating recent developments from the literature. For example, we will take inspiration from *fastText* [8] and incorporate a word embedding lookup table into our architecture. We will also consider adding n-grams as features. Optionally, we may also incorporate morphological information by considering character n-grams [2].

2.1.2 Convolutional Neural Network

Convolutional neural networks (CNNs) have gained a lot attention for text classification in recent years. One way to categorize them is to distinguish between CNNs that operate at the word level and CNNs that operate at the character level.

Kim [10] proposes a rather simple CNN for sentence classification. The model operates on pre-trained word embeddings or word embeddings that are jointly learned with the task. The author also combines the two approaches. Having one convolutional layer with multiple filters of different widths, they perform max-pooling over the entire sentence. The model yields good results on the evaluated datasets, which contain only sentences. Since titles have a similar length, we argue that this model is a good initial choice for the problem at hand. Optionally, this approach could be extended by introducing more layers, varying the pool-size, and introducing Dynamic k-Max Pooling [9].

Optionally, we explore a character level approach presented by Zhang et al. [16] and later improved

upon by Conneau et al [4]. In both approaches, the text is truncated or padded to a fixed size of 1014 characters. After the transformation to character embeddings, the remaining character sequence is propagated through up to 29 convolutional layers. Good results compared to BoW baselines are reported on datasets that have a few million samples. The samples in those datasets have rather medium length. Statistics for some of the datasets are reported by Yang et al. [13]. We will analyze if incorporating a similar architecture boosts the classification performance. However, the truncation length of 1014 characters may be too large or small for the title or full-text datasets, respectively. Also the number and size of layers should be adjusted to the text length. Therefore, we carefully optimize the truncation length as well as number and size of layers for full-text and titles individually on a validation set.

2.1.3 Recurrent Neural Network

For text classification, it is an open question whether CNNs or RNNs are superior [14]. When Zhang et al. [16] introduced their character-based CNN approach, they compared with a fairly simple LSTM approach, which was still able to achieve results that were close to the CNN and the BoW baseline in many cases. On the two largest datasets, it outperforms all BoW methods and most of the CNN architectures. Yang et al. [13] proposed a RNN to learn a sentence representation from words and then use another RNN to learn a document representation. The representation is obtained by taking the weighted average over the RNN's output at each step. The attention mechanism [1] determines the weights. The hierarchical approach shows great improvement on some of the text classification datasets introduced by Zhang et al. However, Conneau et al. [4] note that this approach can only be applied to datasets whose samples contain multiple sentences, which is not always the case.

We will employ an RNN in our research, too. Specifically, we will start with the rather simple LSTM which Zhang et al. employed. Optionally, we will experiment with techniques that are common for LSTMs or have emerged from recent literature. Specifically, we will experiment with other output representation techniques, such as only taking the last output instead of averaging over the output at all time steps as employed by Zhang et al.. For the weighted average, we will employ the attention mechanism. Unfortunately, titles do not consist of multiple sentences, so adopting the same hierarchical approach as Yang et al. is not possible for titles. However, since for a fair comparison of titles and full-text the richer structure of full-text should be acknowledged, we may explore this technique for experiments on full-text. Finally, a very successful recent development in deep learning is dropout [12], a regularization technique for neural networks. Zaremba et al. [15] and Gal et al. [6] explored ways to apply this technique successfully for recurrent neural networks, which allows the training of large, more expressive models. We may incorporate dropout in a similar way into our LSTM.

2.1.4 Addressing Multi-Labeling

The semantic annotation task is formally a *multi-labeling problem*, where instead of belonging to exactly one class, each sample is assigned a set of labels. This is an important difference to most of the previous research that deal with large datasets. *Binary Relevance* is a common approach to adapt a classifier for a multi-labeling problem. This technique trains a separate binary classifier for each label which determines whether to assign the label. In our study, however, we deal with datasets that have thousands of labels. Training a separate deep learning classifier for each label is impractical, especially with such large sample sizes as in our study. Galke et al. [7] had some success with a simplistic approach. A label is assigned if the output layer activated with softmax outputs a value above a certain threshold. We will initially work with the same multi-labeling technique. However, if the previously described neural methods show rather poor performance, we will explore more sophisticated techniques for adapting the algorithms for multi-labling. Therefore, we consider these techniques optional for our research period. For example, Chen et al. [3] propose to use an RNN to predict a sequence of target labels. The feature vector of the base classifier, in their case CNN followed

by a fully-connected layer, is provided as input at every timestep, alongside the output of the previous step. The sequence ends when the RNN outputs a special "END"-token.

2.2 Datasets

In order to simulate a realistic scenario, we build English datasets from several digital libraries. From each library, we obtain one dataset that only contains titles, hereafter called *Title*, and one dataset containing full-texts, called *Full-Text*.

EconBiz[1] is a search portal for economics and business studies. Currently, it contains 2,485,000 English entries, out of which 615,000 are open access. Each entry is annotated by experts with a variable number of subject headings taken from a standardized set, the "STW Thesaurus for Economics"[2].

PubMed Central[3] is an archive of biomedical and life science literature provided by the US National Library of Medicine. It comprises of 4,300,000 articles, which can be accessed freely and which are mostly English. However, only 1,500,000 are open access and therefore allow text mining. The articles are annotated with "Medical Subject Headings" (MeSH)[4].

Optionally, we consider the use of a dataset from the political domain provided by the German Information Network International Relations and Area Studies[5] and explore ways to retrieve a corresponding large title dataset. This could be implemented by crawling metadata from the search portal service provided by International Relations and Area Studies Online[6].

The datasets from the EconBiz portal have already been retrieved by members of ZBW - Leibniz Information Centre for Economics. If we experience unexpected technical or legal issues during the retrieval of the datasets from PubMed Central, we will fall back onto the *New York Times Annotated Corpus Dataset* (NYT), which contains 1,846,656 news articles. The titles of these articles comprise the NYT Title dataset. In order to obtain a scenario where there are more titles than available full-texts, we will randomly select a subset that matches the size of the largest Full-Text dataset from the other domains (between 250,000 and 600,000) and consider this the NYT Full-Text dataset.

2.3 Experiments and Evaluation

In order to answer **RQ2** and **RQ3**, we need to specify $x_1, \ldots, x_k$, i.e. , the factor to determine the sample size of the title data subset. Due to high computational costs when training deep learning models on large amounts of data, an exhaustive examination is impractical. We therefore evaluate on exponentially increasing sample sizes. In particular, we choose $k = 5$ and set $x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 8$. Due to the fact that more training data does not hurt the classification performance, to answer **RQ1**, it is sufficient to evaluate the performance when using the full Title dataset. We set $x_5 = |D_t|/|D_f|$ for that matter.

We may then perform experiments on Title for each x_i, $D \in \mathcal{D}$, and $c \in \mathcal{C}$ and calculate the respective $F_1^c(D_{t_i})$. Moreover, we conduct experiments on Full-Text and calculate $F_1^c(D_f)$. An overview of the experiments is shown in Table 2.3.

Lastly, we evaluate the experiments with respect to our research questions. **RQ1** can be answered directly by checking if $\max_{c \in \mathcal{C}} F_1^c(D_{t_5}) \leq \max_{c' \in \mathcal{C}} F_1^{c'}(D_f)$. For **RQ2** and **RQ3**, we calculate the respective r_i values. We analyze the ratios' growth behaviors by plotting the values (x_i, r_i) in a coordinate system and visually inspecting the resulting graphs. Optionally, we may use regression techniques to determine a good fit for the functional form of the graphs.

2.4 Implementation

Galke et al. [7] released the source code of their study publicly[7], which provides a pipeline for multi-label text classification that can partly be reused for our

[1] https://www.econbiz.de/
[2] http://zbw.eu/stw/version/latest/about
[3] https://www.ncbi.nlm.nih.gov/pmc/
[4] https://www.nlm.nih.gov/mesh/meshhome.html
[5] http://www.fiv-iblk.de/eindex.htm
[6] https://www.ireon-portal.eu/
[7] https://github.com/quadflor/Quadflor

Method	Economics						Medical Science/NYT						Political Science					
Base-MLP	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles
Soph-MLP	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles
CNN	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles
RNN	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles	Full-Text	1×	1.5×	2×	3×	all Titles

Table 3: Experiments conducted for master thesis research when all optional experiments are included. Each cell not in the first row or first column denotes an experiment. $x\times$ denotes the size of Title training set relative to Full-Text training set.

research. Therefore, we will base our implementation on this framework by extending the vectorization step and classification step by the respective models. Furthermore, we will adapt the evaluation scheme to match our experimental setup accordingly.

The neural network classifiers will be implemented with the deep learning frameworks TensorFlow[8] and Keras[9], which we choose due to their rich documentation, large user base, and continuous support.

3 Conclusion

In our research, we will study to which extent the compelling amount of title data available for automatic semantic annotation can improve classification performance in comparison to relatively smaller amounts of full-text data. To this end, we develop and employ deep learning methods, which are specifically known to benefit from large amounts of data. The results will be useful for digital libraries in particular, and the NLP community in general.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[8] `https://www.tensorflow.org/`

[9] `https://keras.io/`

[3] G. Chen, D. Ye, E. Cambria, J. Chen, and Z. Xing. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. IJCNN, 2017.

[4] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*, 2016.

[5] A. M. Dai and Q. V. Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087, 2015.

[6] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027, 2016.

[7] L. Galke, F. Mai, A. Schelten, D. Brunsch, and A. Scherp. Are titles just perfectly good...? a comparison of large-scale multi-label document classification. *arXiv preprint arXiv:1705.05311*, 2017.

[8] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[9] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv:1404.2188*, 2014.

[10] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[11] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz. Large-scale multi-label text

classificationrevisiting neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452. Springer, 2014.

[12] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[13] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.

[14] W. Yin, K. Kann, M. Yu, and H. Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[15] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[16] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.