Hierarchical Multi-label Classification of Text with Capsule Networks

Rami Aly, **Steffen Remus**, and **Chris Biemann**

Language Technology group
Universität Hamburg, Hamburg, Germany
{5aly,remus,biemann}@informatik.uni-hamburg.de

Abstract

Capsule networks have been shown to demonstrate good performance on structured data in the area of visual inference. In this paper we apply and compare simple shallow capsule networks for hierarchical multi-label text classification and show that they can perform superior to other neural networks, such as CNNs and LSTMs, and non-neural network architectures such as SVMs. For our experiments, we use the established Web of Science (WOS) dataset and introduce a new real-world scenario dataset, the BlurbGenreCollection (BGC). Our results confirm the hypothesis that capsule networks are especially advantageous for rare events and structurally diverse categories, which we attribute to their ability to combine latent encoded information.

1 Introduction

In hierarchical multi-label classification (HMC), samples are classified into one or multiple class labels that are organized in a structured label hierarchy (Silla and Freitas, 2011). HMC has been thoroughly researched for traditional classifiers (Sun and Lim, 2001; Silla and Freitas, 2011), but with the increase of available data, the desire for more specific and specialized hierarchies increases. However, since traditional approaches fail to generalize adequately, more sophisticated and robust classification methods are receiving more attention. Complex neural network classifiers on the contrary are computationally expensive, difficult to analyze, and the amount of hyperparameters is significantly higher as compared to other classification approaches. This makes it difficult to apply the *local classifier approach* (Silla and Freitas, 2011), where multiple classifiers are employed to cover different parts of the hierarchy. Therefore, in this paper we focus on the *global approach* – one classifier that is able to capture the entire hierarchy at once. There are indications that capsule networks (Hinton et al., 2011; Sabour et al., 2017) are successful at finding, adapting, and agreeing on latent structures in the underlying data in the area of image recognition as well as recently in the field of natural language processing (Zhao et al., 2018). This insight motivates our research question: To which extent can the capabilities of capsule networks be transferred and applied to HMC in order to capture the categories' underlying structures?

In our experiments[1] we compare HMC-adjusted capsule networks to several baseline neural as well as non-neural architectures on the *BlurbGenreCollection* (BGC), a dataset which we collected and that consists of so-called blurbs of books and their hierarchically structured writing genres. Additionally, we test our hypothesis on the *Web of Science* (WOS) dataset (Kowsari et al., 2017). The main benefit of capsules is their ability to encode information of each category separately by associating each capsule with one category. Combining encoded features independently for each capsule, and thus category, enables capsule networks to handle label combinations better than previous approaches. This property is especially relevant for HMC since documents that for instance only belong to a parent category, e.g. *Fiction*, often share similar features such as the most frequent words or n-grams with documents that additionally classify into one of the parent's child labels, e.g. *Mystery & Suspense* or *Fantasy*. This makes it difficult for traditional classifiers to distinguish between parent and child labels correctly, especially if the specific combination of labels was never observed during training. This paper contributes in two ways: Firstly, we introduce the new openly accessible *BlurbGenreCollection* dataset for the English language. This dataset is created and only minimally adjusted on basis

[1] Code for replicating results: https://github.com/uhh-lt/BlurbGenreCollection-HMC

of a vertical search webpage for books and thus presents a real-world scenario task. Secondly, we thoroughly analyze the properties of capsule networks for HMC. To the best of our knowledge, capsule networks have not yet been applied and tested in the HMC domain.

2 Related Work

Neural networks for HMC: In hierarchical multi-label classification (HMC) samples are assigned one or multiple class labels, which are organized in a structured label hierarchy (Silla and Freitas, 2011). For text classification (TC), we treat a document as a sample and its categories as labels. *Convolutional Neural Networks* (CNNs) and different types of *Recurrent Neural Networks* (RNNs) (Goodfellow et al., 2016; Kim, 2014), most notably long short-term memory units (LSTMs, Hochreiter and Schmidhuber, 1997) have shown to be highly efficient in TC tasks. For HMC, Cerri et al. (2014) use concatenated multi-layer perceptrons (MLP), where each MLP is associated to one level of the class hierarchy. Kowsari et al. (2017) use multiple concatenated deep learning architectures (CNN, LSTM, and MLP) to HMC on a dataset with a rather shallow hierarchy with only two levels. Similar to Kiritchenko et al. (2005), Baker and Korhonen (2017) treat the HMC task as a multi-label classification problem that considers every label in the hierarchy, but they additionally leverage the co-occurrence of labels within the hierarchy to initialize the weights of their CNN's final layer (Kurata et al., 2016).

Capsule Networks: Capsule networks encapsulate features into groups of neurons, so-called capsules (Hinton et al., 2011; Sabour et al., 2017). Originally introduced for a handwritten digit image classification task where each digit has been associated with a capsule, capsules have shown to learn more robust representations for each class as they capture parent-child relationships more accurately. They reached on-par performance with more complex CNN architectures, even outperforming them in several classification tasks such as the *affNIST* and *MultiMNIST* dataset (Sabour et al., 2017). First attempts to use capsules for sentiment analysis were carried out by (Wang et al., 2018) on the basis of an RNN, however, they did not employ the routing algorithm, thus highly limiting the capabilities of capsules. Zhao et al. (2018) show that capsule networks can outperform traditional neural networks for TC by a great margin when training on single-labeled and testing on multi-labeled documents of the Reuters-21578 dataset since the routing of capsules behaves like a parallel attention mechanism regarding the selection of categories. By connecting a BiLSTM to a capsule network for relation extraction, Zhang et al. (2018) show that capsule networks improve at extracting n-ary relations, with $n > 2$, per sentence and thus confirm the observation of (Zhao et al., 2018) in a different context. For multi-task learning, Xiao et al. (2018) use capsule networks to improve the differentiation between tasks. They encapsulate features in different capsules and use the routing algorithm to cluster features for each task. Further applications to NLP span aggression, toxicity and emotion detection (Srivastava et al., 2018; Rathnayaka et al., 2018), embedding creation for knowledge graph completion (Nguyen et al., 2019), and knowledge transfer of user intents (Xia et al., 2018). Despite the suitable properties of capsule networks to classify into hierarchical structured categories, they have not yet been applied to HMC. This work aims to fill the gap by applying and thoroughly analyzing capsules' properties at HMC.

3 Capsule Network for HMC

For each category in the hierarchy, an associated capsule outputs latent information of the category in form of a vector as opposed to a single scalar value used in traditional neural networks. The vector is equivariant with its length defining the pseudo-probability of its activation and its orientation representing different cases of a category's existence. This distributional representation in the form of a vector instead of a scalar makes capsules exponentially more informative than traditional perceptrons (Sabour et al., 2017).

The input of capsules in the first capsule layer of a capsule network is called *primary capsules* and can be of arbitrary dimension, typically coming from a convolutional layer or from the hidden state of a recurrent network. The output vector of a primary capsule represents latent information such as local order and semantic representations of words (Zhao et al., 2018). Each capsule j in the next layer, called *classification capsules*, take as input the weighted sum $\boldsymbol{s}_j = \sum_i c_{j|i}\hat{\boldsymbol{u}}_{j|i}$ of the prediction vectors of all primary capsules

i. A capsule's prediction vector $\hat{\boldsymbol{u}}_{j|i}$ is generated by multiplying the output $\boldsymbol{u}_{j|i}$ by a weight matrix W_{ij}. Since the length of a vector of a classification capsule should be interpreted as the probability of the corresponding category, a squashing function $\boldsymbol{v}_j = squash(\boldsymbol{s}_j)$ is applied, which scales the output of each classification capsule non-linearly between zero and one. The coupling coefficients $c_{j|i}$ that determine the contribution of each primary capsule's output to a classification capsule are calculated using a dynamic routing heuristic (Sabour et al., 2017). It iteratively decides the routes of capsules and thus how to cluster features for each category. The pseudocode for the full routing algorithm is written in Algorithm 1.

```
Routing algorithm
Result: $\boldsymbol{v}_j$
Initialization: $\forall i \in Primary.\forall j \in Classification : b_{j|i} \leftarrow 0.$
for r iterations do
    $\forall i \in Primary : \boldsymbol{c}_i \leftarrow \text{softmax}(\boldsymbol{b}_i)$
    $\forall j \in Clas. : \boldsymbol{v}_j \leftarrow \text{squash}(\sum_i c_{j|i} \hat{\boldsymbol{u}}_{j|i})$
    $\forall i \in Primary.\forall j \in Clas. : b_{j|i} \leftarrow b_{j|i} + \hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$
end
```

Algorithm 1: Routing algorithm as described in (Sabour et al., 2017)

The coupling coefficients are generated by applying the softmax function to the log prior probabilities that primary capsule i should be coupled to classification capsule j. The probability is higher when the primary capsule's prediction vector is more similar to the classification capsule's output. Therefore, primary capsules try to predict the output of the capsule in the subsequent layer. Since $\boldsymbol{v}_j$ is partially determined by $\boldsymbol{u}_{j|i}$, their similarity increases for the next iteration. Thus a convergence is guaranteed.

This routing algorithm is superior regarding its ability to combine and generalize information compared to primitive routing algorithms such as max-pooling layers, as the latter only stores the most prominent features while the others are ignored. This leads to CNNs having more difficulty differentiating between classes with highly similar features (Sabour et al., 2017), but since most label combinations appear rarely and categories often share features with their parents, it is a desirable property to exploit for hierarchical classification.

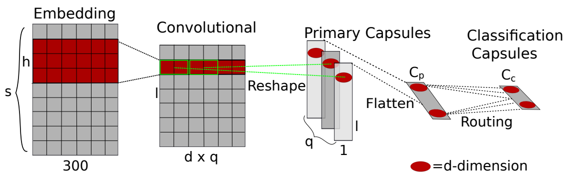


Figure 1: Architecture of our capsule network with d being the dimensionality of a capsule's output.

Architecture: The HMC task is converted to a multi-label classification task using the hierarchy of labels: All explicitly labeled classes must also include all ancestor labels of the hierarchy. The architecture of our capsule network is shown in Figure 1 and consists of four layers. We designed a minimal capsule network, similar to CapsNet-1 in (Xiao et al., 2018) in order to benefit from capsules and dynamic routing while maintaining high comparability to a similarly simple CNN. In our network, the primary capsules take as input the output created by a preceding convolutional layer. For each classification capsule, the routing algorithm is then used to cluster the outputs of all c_p primary capsules. The pseudo-probability $||\boldsymbol{v}_j||$ is then assigned to the category associated with the effective classification capsule. We follow Sabour et al. (2017), and use their *margin loss* function.

Leveraging Label Co-occurrence: We further follow the layer weight initialization introduced by (Kurata et al., 2016) in order to leverage label co-occurrences during the learning process of a neural network. Since label co-occurrences such as {*Fiction*, *Mystery & Suspense*} or {*Fiction*, *Fantasy*} naturally occur in HMC because of parent-child relationships between categories, we aim to bias the learning process of the capsule network in respect to the co-occurrences in the dataset by initializing W with label co-occurrences. Weights between a primary capsule and the co-occurring classification capsules are initialized using a uniform distribution while all other values are set to zero.

Label Correction: A classifier may assign labels to classes that do not conform with the underlying hierarchy of the categories as the activation function as well as the routing algorithm look at each category separately. For instance, if the capsule network only assigns the label *Fantasy* then the prediction is inconsistent with the hierarchy as its parent *Fiction* has not been labeled. Inconsistencies with respect to the hierarchical structure of categories are corrected by a post-processing step.

We applied three different ways of label correction: Correction by *extension*, *removal* and *threshold*. The former two systematically add parent or remove parentless labels to make the prediction consistent (Baker and Korhonen, 2017). Therefore, the first method adds *Fiction* to the predictions while the second one removes the prediction *Fantasy* (and all its children) in its entirety. Correction by threshold calculates the average confidence of all ancestors for an inconsistent prediction and adds them if above the threshold (Kiritchenko et al., 2005).

4 Experiments

Datasets: We test our hypothesis on two different datasets with fundamentally different properties, the BlurbGenreCollection[2] (BGC), and the WOS-11967 (Web of Science, Kowsari et al., 2017).

The BGC dataset consists of book blurbs (short advertising texts) and several book-related meta-information such as author, date of publication, number of pages, and so on. Each blurb is categorized into one or multiple categories in a hierarchy. With their permission, we crawled the Penguin Random House website and performed cleaning steps, such as: removing categories that do not rely on content (e.g. audiobooks), and removing category combinations that appear less than five times. The dataset follows the well-known dataset properties as described in (Lewis et al., 2004): Firstly, at least one writing-genre is assigned to each book and secondly, every ancestor of a book's label is assigned to it as well. It is important to note that the most specific genre of a book does not have to be a leaf. For instance, the most specific category of a book could be *Children's Books*, although Children's Books has further sub-genres, such as *Middle Grade books*. Furthermore, in this dataset, each child-label has exactly one parent, forming all-together a hierarchy in form of a forest. Nonetheless, the label distribution remains highly unbalanced and diverse, with a total of $1,342$ different label co-occurrences from a pool of 146 different labels arranged on 4 hierarchy levels.

The WOS dataset consists of abstracts of published papers from the Web of Science. The hierarchy of the WOS dataset is shallower, but significantly broader, with fewer classes in total. In addition to having only as many co-occurrences as leaf nodes, measuring the entropy of label combinations shows that the dataset is unnaturally balanced – a consequence of the dataset's requirement to assign exactly two labels to each example. Table 1 shows further important quantitative characteristics of both datasets.

	BGC	WOS-11967
Number of texts	91,892	11,967
Average number of tokens	93.56	125.90
Total number of classes	146	40
Classes on level 1;2;3;4	7; 46; 77; 16	7; 33; -; -
Average number of labels	3.01	2
Total number of label co-occurrences	1342	33
Co-occurrence entropy (normalized)	0.7345	0.9973
Samples per category standard deviation	4374.19	529.43

Table 1: Quantitative characteristics of both datasets. Normalized entropy is the quotient between entropy and the log of co-occurrence cardinality.

Feature selection: Since CNNs and our capsule network require a fixed input length, we limit the texts to the first 100 tokens, which covers the complete input for over 90% of the dataset. We remove stop-words, most punctuation and low-frequency words (< 2). For the BGC, we kept special characters like exclamation marks as they can be frequently found in blurbs that have a younger target audience and hence could provide useful information. We are using pre-trained fastText embeddings[3] provided by Bojanowski et al. (2017) and adjust them during training.

Baselines: We employ a one-vs-rest classification strategy using one SVM (Cortes and Vapnik, 1995) for each label with linear kernels and tf-idf values in a bag-of-words fashion as feature vectors. Also, we apply the CNN as described by Kim (2014) and an LSTM with recurrent dropout (Gal and Ghahramani, 2016).[4] For all experiments we use the initialization strategy as described in (Baker and Korhonen, 2017), which takes label co-occurrences for initializing the weights of the final layer, and the label correction method by thresh-

[2] The dataset is available at https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html

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

[4] All neural networks use the Adam optimizer, a dropout probability of 0.5 and a minibatch size of 32. LSTM and CNN use the binary cross entropy loss. Further hyperparameters for (BGC, WOS) – CNN: filters: (1500, 1000), windows: $\{3,4,5\}$, l. rate: (0.0005, 0.001), l. decay: (0.9, 1), epochs: (30, 20); LSTM: hidden units: (1500, 1000), l. rate: (0.005, 0.001), epochs: (15, 25); capsule network: num. capsules: (55, 32), windows: (90, 50), primary/class. cap. dim.: 8/16, l. rate: (0.001, 0.002), l. decay: (0.4, 0.95), epochs: 4

	BGC				WOS-11967			
Model	Recall	Precision	F_1	Subset Acc.	Recall	Precision	F_1	Subset Acc.
SVM	61.11	**85.37**	71.23	35.79	72.43	89.84	80.20	56.47
CNN	64.75 ± 0.41	83.87 ± 0.09	73.08 ± 0.27	37.26 ± 0.52	$\mathbf{84.06 \pm 0.93}$	$\mathbf{91.68 \pm 1.00}$	$\mathbf{87.71 \pm 0.58}$	75.16 ± 1.66
LSTM	69.12 ± 1.24	75.49 ± 3.54	72.16 ± 1.01	$\mathbf{37.99 \pm 1.52}$	83.78 ± 1.69	87.56 ± 1.04	85.63 ± 1.22	$\mathbf{76.80 \pm 2.15}$
Caps. Network	$\mathbf{71.73 \pm 0.63}$	77.21 ± 0.54	$\mathbf{74.37 \pm 0.35}$	37.70 ± 0.68	80.67 ± 1.27	82.75 ± 2.42	81.69 ± 0.70	64.97 ± 0.49

Table 2: All results with their corresponding 95% confidence intervals, measured across three runs.

old with a confidence value of 0.2.[5] The dataset is split into 64% train, 16% validation and 20% test. For evaluation, we measure subset accuracy, micro-averaged recall, precision, and F_1 as defined in (Sorower, 2010; Silla and Freitas, 2011).

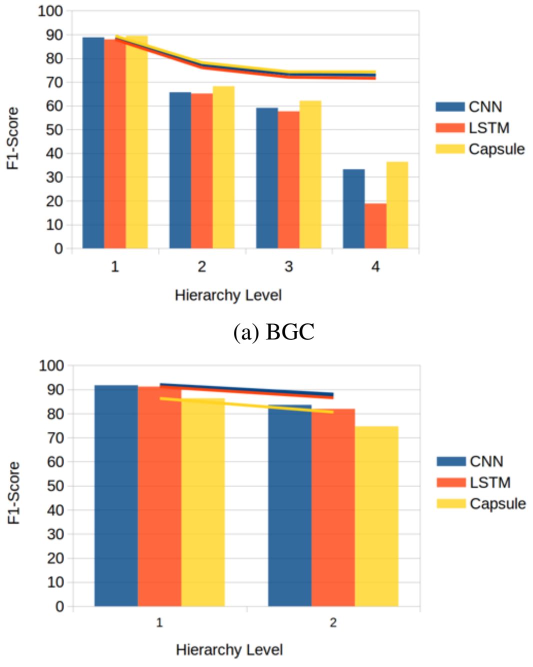
5 Results

Results are shown in Table 2. Regarding the BGC dataset, the capsule network yields the highest F_1 and recall, the SVM the highest precision, while the LSTM showed the best result in subset accuracy. On WOS, all neural network architectures beat the baseline SVM model by a substantial margin. However, both, the SVM and the capsule network, are substantially outperformed by the CNN and LSTM. In Figure 2 we further observe a performance decline for deeper levels of the hierarchy. On BGC, the capsule network performs best on every level of the hierarchy with an increasing margin for more specific labels.

5.1 Identification of label co-occurrences

We argue that the pronounced performance difference between the datasets is due to the ability of capsules to handle label combinations better than the CNN and LSTM. We observe, as shown in Figure 4, that capsule networks are beneficial for examples with many label assignments. While the capsule network performs worse on BGC for a label set cardinality of 1 and 2, it starts to perform better at a cardinality of 3 and almost doubles the F_1 of all baselines for 9 and 11. The number of examples decreases exponentially with the label set cardinality, so that the ability of networks to combine labels is becoming increasingly important.

In contrast, in the WOS dataset, exactly one parent-child label combination is assigned to each example, resulting in a label set cardinality of two for the whole dataset. There are comparably few label combinations, which occur with a high frequency in the dataset (cf. Table 1). The benefit of capsules can thus not apply here.



(a) BGC

(b) WOS

Figure 2: Scores on different levels for the BGC (a) and WOS (b). The lines are the cumulative scores.

To verify this hypothesis, we conduct a further test exclusively on BGC examples with label combinations that have not been observed during training (5,943 samples). As shown in Table 3, the capsule network again achieves the highest F_1 score, outperforming the other networks, especially in terms of recall. In order to create hierarchical inconsistencies in the WOS dataset, we test two modifications on the training data while the test data is kept the same: *a*) 50% of all child labels are removed, and *b*) for each sample, either the child or the parent label is kept. Results of this study are shown in Table 4. Removing 50% of the children labels results in the capsule network being more similar to the CNN and LSTM in terms of subset accuracy. However, for the second modification, where label combinations are completely omitted for training, the capsule network significantly outperforms both networks. Figure 3 shows that different primary capsules are routed to the classification capsule representing the parent cat-

[5] These options consistently performed well in preliminary experiments.

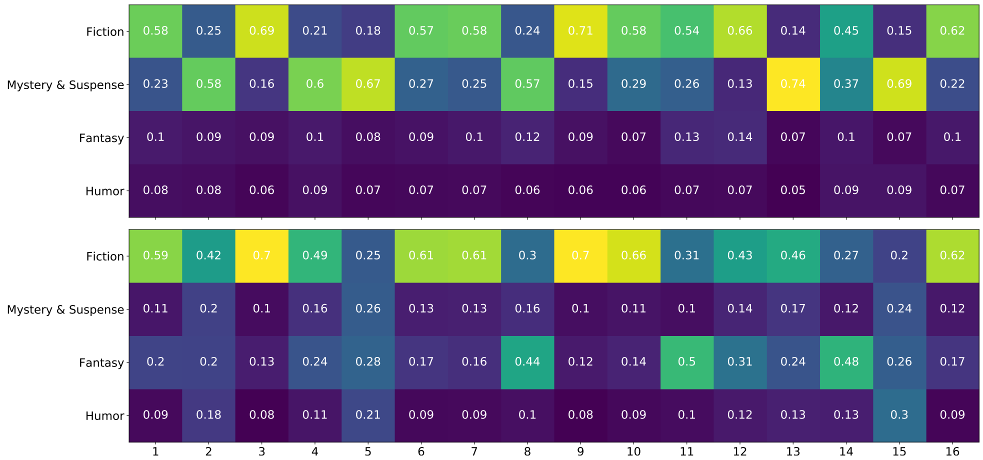


Figure 3: Connection strength between primary capsules (x-axis) and classification capsules (y-axis) for two BGC samples: top belonging to {*Fiction*, *Mystery & Suspense*} and bottom to {*Fiction*, *Fantasy*} with *Fiction* being their parent category. A reduced number of primary capsules and categories was used for visualization purposes.

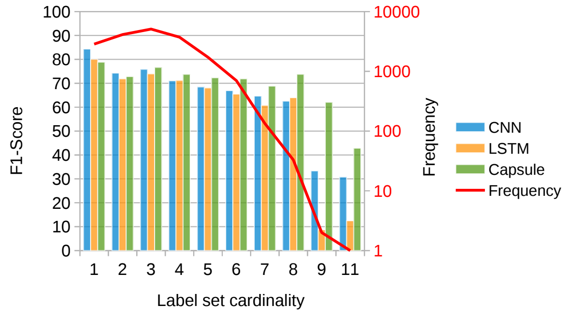


Figure 4: Test F_1-scores of classifiers for different label cardinalities.

BGC, unobserved	R	P	F_1
CNN	46.21	**68.95**	55.34
LSTM	45.79	60.48	52.13
Capsule Net.	**53.30**	61.21	**56.98**

Table 3: Performance results on the test set with label combinations not seen during training.

egory *Fiction* than to the children. Some primary capsules learn features for specific children categories. For instance Primary Capsule 5 is not inclined to any category for the bottom sample because of missing features for *Mystery & Suspense* in this sample. Some capsules distribute their connection strength to the parent and child category evenly, likely due to the categories' similarities. To combine encoded features for each category separately while using the softmax to ensure that primary capsules encapsulate features of specific categories appears to be the main cause of these significant performance differences. These observations also align with previous work, especially see (Sabour et al., 2017; Zhao et al., 2018).

Modified WOS	50% Child Labels		Either Parent or Child	
	F_1	Acc	F_1	Acc
CNN	**75.15**	**36.28**	41.93	16.36
LSTM	73.00	35.09	38.74	5.28
Capsule Net.	71.59	35.21	**67.23**	**34.27**

Table 4: Results on the modified WOS training data. Firstly, by removing 50% of the children labels and secondly, by removing label combinations completely.

6 Conclusion

This first application of capsule networks to the HMC task indicates that the beneficial properties of capsules can be successfully utilized. By associating each category in the hierarchy with a separate capsule, as well as using a routing algorithm to combine in capsules encoded features, capsule networks have shown to identify and combine categories with similar features more accurately than the baselines. The introduced dataset, the BlurbGenreCollection (BGC), is compiled from a real-world scenario and is indicative of the promising properties of capsule networks for HMC tasks, since most hierarchically organized datasets consist of substantial amounts of rare label combinations, where algorithms are very likely to be confronted with unseen label combinations.

This initial attempt shows the advantage of simplistic capsule networks over traditional methods for HMC. Future architectures could for example employ a cascade of capsule layers with each capsule in one layer being associated to a category of one specific level in the hierarchy.

References

Simon Baker and Anna Korhonen. 2017. Initializing neural networks for hierarchical multi-label text classification. In *Proceedings of the 16th Biomedical Natural Language Processing Workshop*, pages 307–315, Vancouver, Canada.

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

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

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

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems 2016*, pages 1019–1027, Barcelona, Spain.

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

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. 2011. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51, Espoo, Finland.

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

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

Svetlana Kiritchenko, Stan Matwin, and A. Fazel Famili. 2005. Functional annotation of genes using hierarchical text categorization. In *BioLINK SIG: Linking Literature, Information and Knowledge for Biology*, Detroit, MI, USA. Workshop track.

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

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Improved neural network-based multi-label classification with better initialization leveraging label cooccurrence. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 521–526, New Orleans, LA, USA.

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

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA.

Prabod Rathnayaka, Supun Abeysinghe, Chamod Samarajeewa, Isura Manchanayake, and Malaka Walpola. 2018. Sentylic at IEST 2018: Gated recurrent neural network and capsule network based approach for implicit emotion detection. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 254–259, Brussels, Belgium.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30*, pages 3856–3866, Long Beach, CA, USA.

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

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

Saurabh Srivastava, Prerna Khurana, and Vartika Tewari. 2018. Identifying aggression and toxicity in comments using capsule network. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 98–105, Santa Fe, NM, USA.

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

Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment analysis by capsules. In *Proceedings of the 2018 World Wide Web Conference*, pages 1165–1174, Lyon, France.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099, Brussels, Belgium.

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

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018. Attention-based capsule networks with dynamic routing for relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, page 986–992, Brussels, Belgium.

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