Cancer Hallmark Text Classification Using Convolutional Neural Networks

Simon Baker [1,2] **Anna Korhonen** [2] **Sampo Pyysalo** [2]
[1]Computer Laboratory, 15 JJ Thomson Avenue
[2] Language Technology Lab, DTAL
University of Cambridge, UK
simon.baker@cl.cam.ac.uk, alk23@cam.ac.uk, sampo@pyysalo.net

Abstract

Methods based on deep learning approaches have recently achieved state-of-the-art performance in a range of machine learning tasks and are increasingly applied to natural language processing (NLP). Despite strong results in various established NLP tasks involving general domain texts, there is only limited work applying these models to biomedical NLP. In this paper, we consider a Convolutional Neural Network (CNN) approach to biomedical text classification. Evaluation using a recently introduced cancer domain dataset involving the categorization of documents according to the well-established hallmarks of cancer shows that a basic CNN model can achieve a level of performance competitive with a Support Vector Machine (SVM) trained using complex manually engineered features optimized to the task. We further show that simple modifications to the CNN hyperparameters, initialization, and training process allow the model to notably outperform the SVM, establishing a new state of the art result at this task. We make all of the resources and tools introduced in this study available under open licenses from https://cambridgeltl.github.io/cancer-hallmark-cnn/ .

1 Introduction

A major goal of cancer research is to understand the biological mechanisms involved in tumorous growths starting in the body, being sustained, and turning malignant. Cancer is often described in the biomedical literature by its *hallmarks*; a set of interrelated biological properties and behaviors that enable cancer to thrive in the body. The hallmarks of cancer were first introduced in the seminal paper of Hanahan and Weinberg (2000), the most cited paper in the journal *Cell*. The paper introduces six hallmarks, which were then extended in a follow-up paper (Hanahan and Weinberg, 2011) by another four, forming the set of ten hallmarks that are known today. The current set of hallmarks distill our knowledge of the disease into a fixed set of alterations in cell physiology that affect malignant growth, such as self-sufficiency in growth signals, insensitivity to growth-inhibitors, evasion of programmed cell death, limitless replicative potential, sustained angiogenesis, and tissue invasion.

In the context of biomedical text mining, the original six hallmarks of cancer were used as an organizing principle in the BioNLP Shared Task 2013 Cancer Genetics task (Pyysalo et al., 2013b), which involved the extraction of events (biological processes) from cancer domain texts. The hallmarks have also inspired other information extraction efforts and the development of tools such as *OncoSearch* (Lee, 2014) and *OncoCL* (Doland, 2014). In recent work, Baker et al. (2016) introduced a corpus comprised of over 1,800 abstracts from biomedical publications annotated with the ten hallmarks of cancer. Baker et al. also proposed a machine learning based method for classifying text according to the hallmarks. The approach utilizes a conventional NLP pipeline that extracts a feature-rich representation that is used to train support vector machine (SVM) classifiers. The method achieves a respectable level of performance, identifying hallmarks with an average F-score of 77%, but with the cost of involving a lengthy and computationally demanding NLP pipeline.

Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016), pages 1–9, Osaka, Japan, December 12th 2016.

In this work, our focus is on studying biomedical text classification using machine learning methods that emphasize *feature learning* rather than manual feature engineering. We adopt the task setting and dataset of Baker et al. (2016), but instead of SVMs, we focus on convolutional neural networks (CNN). CNNs were first proposed for image processing (LeCun and Bengio, 1995) and have been recently shown to achieve state-of-the-art performance in a range of NLP tasks, in particular in text classification (Zhang et al., 2015; Severyn and Moschitti, 2015; Zhang and Wallace, 2015). While neural network-based methods in general and "deep" networks in particular are increasingly popular for general domain NLP, there has been comparatively little work applying this class of methods to biomedical text. One recent study applying a CNN model to biomedical text classification task was presented by (Limsopatham and Collier, 2016), who applied CNNs to the task of adverse drug reaction detection in social media messages (Ginn et al., 2014). In addition to the specific subdomain of the source texts and the novel categories represented by the hallmarks of cancer, one factor that sets apart the task here from this previous work is the length of the texts: instead of sentences or brief social media messages, our task involves the classification of publication abstracts typically consisting of hundreds of words.

2 Data

For training and evaluating our methods, we use the corpus of 1852 biomedical publication abstracts annotated for the hallmarks of cancer by Baker et al. (2016). Each abstract in the dataset may be labeled with zero or more of the ten hallmarks, i.e. the task is multi-label classification. The ten hallmarks are summarized below:

Sustaining proliferative signaling: Healthy cells require molecules that act as signals for them to grow and divide. Cancer cells, on the other hand, are able to grow without these external signals.
Evading growth suppressors: Cells have processes that halt growth and division. In cancer cells, these processes are altered so that they don't effectively prevent cell division.
Resisting cell death: Apoptosis is a mechanism by which cells are programmed to die in the event that they become damaged. Cancer cells are able to bypass these mechanisms.
Enabling replicative immortality: Non-cancer cells die after a certain number of divisions. Cancer cells, however, are capable of indefinite growth and division (immortality).
Inducing angiogenesis: Cancer cells are able to initiate angiogenesis, the process by which new blood vessels are formed, thus ensuring the supply of oxygen and other nutrients.
Activating invasion & metastasis: Cancer cells can break away from their site of origin to invade surrounding tissue and spread to distant body parts.
Genome instability & mutation: Cancer cells generally have severe chromosomal abnormalities, which worsen as the disease progresses.
Tumor-promoting inflammation: Inflammation affects the microenvironment surrounding tumors, contributing to the proliferation, survival and metastasis of cancer cells.
Deregulating cellular energetics: Most cancer cells use abnormal metabolic pathways to generate energy, e.g. exhibiting glucose fermentation even when enough oxygen is present to properly respire.
Avoiding immune destruction: Cancer cells are invisible to the immune system.

We divide the dataset into ten binary-labeled datasets (one per hallmark), where the positive examples in each are the abstracts annotated with the hallmark, and negative examples are those that are not. While we generally aim to follow the experimental setup of Baker et al., we chose to split the annotated data into training, development and test subsets instead of applying the cross-validation setup of the study introducing the dataset. Cross-validation setups using all available data fail to make a clear separation between data used for method development and blind data held out for final testing only, and should be avoided in studies involving experimentally driven model refinement (as we do here). Consequently, we initially split the corpus in 70/10/20% proportion to train, development and test sets with a random sampling strategy that aimed to roughly preserve the overall class distribution in each sample. The test set was held out during development and only used in the final experiments. Table 1 shows the distribution of positive and negative labels for each hallmark.

	Train		Devel		Test		Total	
Hallmark	**pos**	**neg**	**pos**	**neg**	**pos**	**neg**	**pos**	**neg**
Sustaining proliferative signaling	328	975	43	140	91	275	462	1390
Evading growth suppressors	172	1131	22	161	46	320	240	1612
Resisting cell death	303	1000	42	141	84	282	429	1423
Enabling replicative immortality	81	1222	11	172	23	343	115	1737
Inducing angiogenesis	99	1204	13	170	31	335	143	1709
Activating invasion and metastasis	208	1095	29	154	54	312	291	1561
Genomic instability and mutation	227	1076	38	145	68	298	333	1519
Tumor promoting inflammation	169	1134	24	159	47	319	240	1612
Cellular energetics	74	1229	10	173	21	345	105	1747
Avoiding immune destruction	77	1226	10	173	21	345	108	1744

Table 1: Annotation statistics

3 Methods

We implement and evaluate two SVM-based methods and two CNN variants, described in the following. All of these machine learning methods are applied to the multi-label task by training ten binary classifiers, one for each hallmark label.

3.1 SVM with Bag of Words Features

We implement a simple classifier using only Bag of Words (BoW) features as a basic SVM baseline. In the BoW approach each document is represented by the set of words appearing in it, discarding word order and frequency information. For training the model, we use the linear kernel SVM implemented in the Scikit-learn (Pedregosa et al., 2011) toolkit. We fine-tune the regularization hyperparameter c conventionally using evaluation on the development dataset with a search between 10^{-2} and 10^{2} on a log scale.

3.2 SVM with Rich Features

For our primary point of reference, we replicated the NLP pipeline and SVM model of Baker et al. (2016) for hallmark classification. This model uses a rich set of features derived from the application of several state-of-the-art systems for biomedical NLP, summarized briefly in the following.[1]

Lemmatized bag of words All non-stop words in the documents are lemmatized using *BioLemmatizer* (Liu et al., 2012) and included as features using a BoW-style representation.
Noun bigrams Compound nouns (without lemmatization) are combined to generate bigram features. Nouns pairs often represent specific, discriminative concepts such as "*gene mutation*".
Grammatical relations triples The *C&C Parser* with a biomedical domain model (Rimell and Clark, 2009) is used to parse the documents, and the *dobj* (direct object), *ncsubj* (non-clausal subject) and *iobj* (indirect object) relations, and their head and dependent words then represented as features.
Verb classes The hierarchical classification of 399 verbs of Sun and Korhonen (2009) is used to generate features for verbs, utilizing all three levels of abstraction by allocating three bits in the feature representation for each concrete class, i.e. one bit for each level of the verb class hierarchy.
Named entities (NE) The *ABNER* NER tool (Settles, 2005) is used to identify five named entity types that are particularly relevant to cancer research: proteins, DNA, RNA, cell lines and cell types. Features are then created pairing each entity type and its associated words.
Medical subject headings (MeSH) The MeSH headings assigned to the documents in the biomedical publication indexing process are included as features using a bag-of-headings representation.
Chemical lists Similarly to MeSH terms, many documents are indexed with chemical identifiers. These identifiers are used analogously to the MeSH terms to generate features.

[1] We refer to Baker et al. (2016) for the further details on this feature representation.

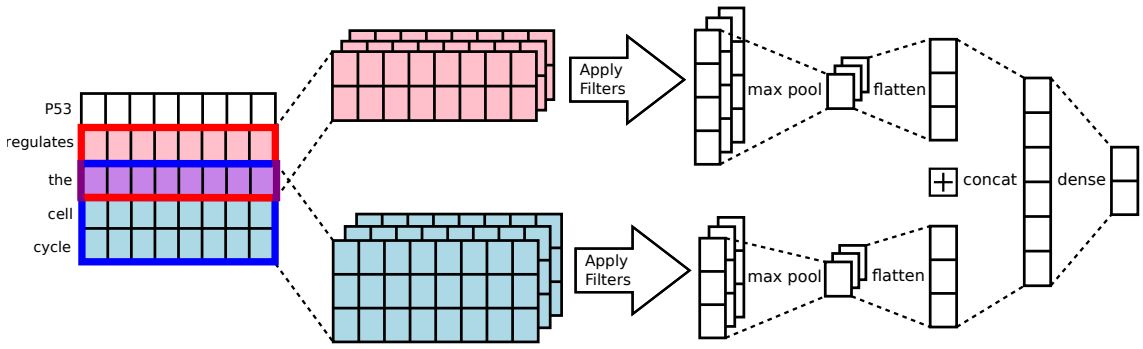

Figure 1: Network architecture

All features are extracted from the training data and are then filtered by frequency to remove features that are too common or too rare, leaving behind only the most discriminating features. We use a linear kernel and fine-tune the regularization parameter c on the development dataset using the same process applied for the BoW model. As there are significantly more negatively labelled documents than positives, we use inverse class weighting in order to correct for the class imbalance when training the classifiers.

3.3 Convolutional Neural Network

We base our CNN architecture on the simple model of Kim (2014). In brief, this model consists of an initial embedding layer that maps input texts into matrices, followed by convolutions of different filter sizes and 1-max pooling, and finally a fully connected layer. The architecture is illustrated in Figure 1. We implemented the neural network using Keras (Chollet, 2015). Model hyperparameters and the training setup were initially based on those applied by Kim (2014), summarized in the following:

Parameter	Value
Word vector size	300 (Google News vectors)
Filter sizes	3, 4, and 5
Number of filters	300 (100 of each size)
Dropout probability	0.5
Minibatch size	50

Table 2: Kim (2014) model parameters

Some of these parameters were further refined in experiments using only the training and development portions of the data (see Section 4.1). In the final test set experiments, we evaluate the network using both the set of parameters used by Kim (2014) as well as with those selected in our development set experiments. We train the models for 20 epochs using categorical cross-entropy loss and the Adam optimization method (Kingma and Ba, 2014).

For regularization, we only apply dropout (Srivastava et al., 2014) before the output layer. We also considered L_2 regularization but did not find a consistent improvement in preliminary experiments.

3.4 Word embeddings

The first layer of the CNN involves mapping words in the input to dense, low-dimensional vectors. These word embeddings are critically important as they represent the "meaning" of the words in the model, e.g. how similar one word is to another. Although it is possible to learn these embeddings from scratch (i.e. random initialization) during the normal training process, recent studies have shown that it is effective to use embeddings that have been separately induced on large, unannotated corpora (Collobert et al., 2011; Kim, 2014). Work in biomedical NLP has further established that word embeddings are domain-dependent: to get the maximal benefit from using pre-trained embeddings for biomedical NLP tasks, the embeddings must be induced using biomedical texts (Stenetorp et al., 2012).

We consider a variety of word embeddings induced using models implemented in the popular `word2vec` package (Mikolov et al., 2013a). First, we use the general-domain Google News vectors

	Source texts		Vectors			
Name	**domain**	**size**	**words**	**dim**	**OOV**	**Reference**
Google News	General	100B	3M	300	31.0%	(Mikolov et al., 2013b)
Pyysalo PM	Bio	3B	2.3M	200	0.52%	(Pyysalo et al., 2013a)
Pyysalo PMC	Bio	2.5B	2.5M	200	0.51%	(Pyysalo et al., 2013a)
Pyysalo PM+PMC	Bio	5.5B	4M	200	0.49%	(Pyysalo et al., 2013a)
Pyysalo Wiki+PM+PMC	General and bio	7.5B	5.4M	200	0.53%	(Pyysalo et al., 2013a)
Chiu win-2	Bio	2.7B	2.2M	200	0.49%	(Chiu et al., 2016)
Chiu win-30	Bio	2.7B	2.2M	200	0.49%	(Chiu et al., 2016)

Table 3: Word vectors

also applied by Kim (2014).[2]. Second, we evaluate three sets of word vectors induced on various combinations of PubMed (PM), PMC and Wikipedia texts by Pyysalo et al. (2013a).[3] Finally, we consider two variants of PubMed-based vectors introduced by Chiu et al. (2016).[4] The properties of these word vectors are detailed in Table 3. Note that unlike the other properties, the out-of-vocabulary rate (OOV) is not a characteristic of the word vectors alone, but the ratio of words in the task training data that do not appear in the work vectors. The high OOV rate for the Google News vectors is due primarily to removal of stopwords, punctuation, and numbers (see also Section 4.1).

3.5 Experimental Setup

Classifier performance is evaluated using the standard precision, recall, and F-score metrics as well as with the area under the receiver operating characteristic curve (AUC). Unlike precision and F-score, AUC is invariant to the positive/negative class distribution. AUC is also more sensitive in summarizing performance over all possible classification thresholds and eliminates the need to pick a specific threshold for evaluation. AUC is therefore recommended for evaluating imbalanced datasets (Zhang and Wallace, 2015). As the dataset is comparatively small and the number of positive examples in particular is very limited for many labels, the random factors in CNN initialization and training can have a substantial effect on the resulting model. To address this issue, we systematically repeated each CNN experiment 10 times and report the mean of the evaluation results.[5] To address overfitting in the CNN, we apply a form of early stopping, testing only the model that achieved the highest results on the development set. In the development experiments, we correspondingly report the highest f-score and AUC from any epoch.

4 Results

In the following, we first summarize results from adapting the basic CNN to the task using the development data, and then present the comparative results on the test set.

4.1 Development results

We considered a range of modifications to the basic CNN model to better adapt it to biomedical domain text classification in general and the specific task studied in this work in particular. Of these modifications, evaluation on the development set identified three that appeared to have beneficial effects on performance: oversampling to address the class imbalance, using in-domain word vectors, and adjusting the filter sizes to the task. We next briefly describe these modifications and the associated results.

Oversampling The dataset is highly biased, with negative examples outnumbering positives more than 10-fold for a number of the labels (Table 1). Standard training on such data is likely to result in models with high precision, low recall, and thus comparatively low F-scores. Addressing this, we oversample the positive examples in the training set with replacement so that their number matches that of the negatives. This modification increased the average F-score on the development set from 85.3% to 86.1%. As expected, the effect on the distribution-independent AUC metric was more limited, improving from 97.3% to 97.5% with oversampling.

[2]Available from `https://code.google.com/archive/p/word2vec/`

[3]Available from `http://bio.nlplab.org/`

[4]Available from `https://github.com/cambridgeltl/BioNLP-2016`

[5]As SVM optimization is convex, repetitions are unnecessary.

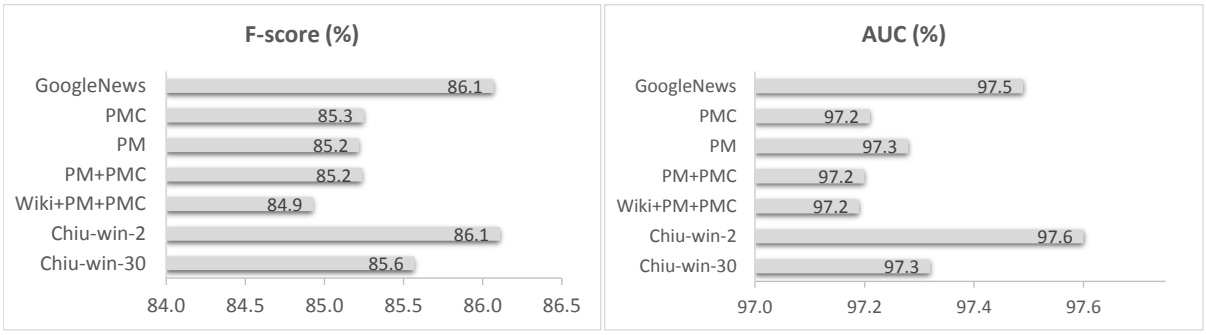

Figure 2: Embedding performance (macro-averaged) on the development dataset

Word embeddings As discussed in Section 3.4, the word vectors used to initialize the embedding layer of the network can have a significant effect on performance. We trained the models using each of the word vectors shown in Table 3 with oversampling (see above) and evaluated development set performance using the maximum F-score and AUC metrics. The results are summarized in Figure 2. Surprisingly, we find that the general domain Google News vectors give very competitive performance despite their high out-of-vocabulary rate (see Table 3), outperforming all in-domain vectors with the exception of the window size 2 word vectors of Chiu et al. (2016). Even these biomedical word vectors only show very modest advantage over the Google News vectors for AUC. In the last development set experiments below and the final test set experiments, we apply the PubMed-based vectors induced with window size 2 from Chiu et al. that were shown to give the best results here.

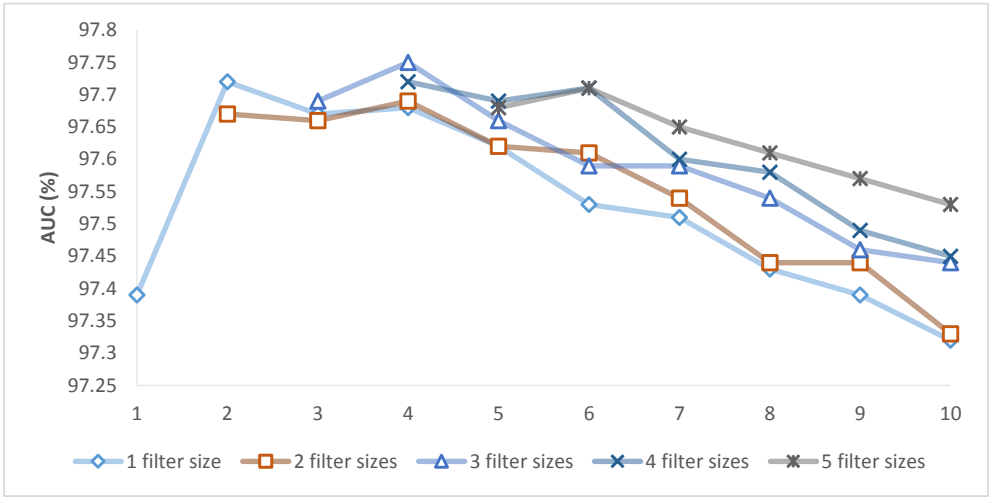

Figure 3: Macro-average AUC with respect to a varying number of filter sizes. Each point on the graph represents the maximum size of filter used (e.g. for *2 filter sizes*, performance with filters of sizes 2 and 3 is plotted at 3).

Filter sizes We experiment with varying the number of filter sizes in the convolutions. The base model of Kim (2014) uses three filter sizes (3,4,5); as part of our hyperparameter search, we investigated what happens to the performance (AUC) with respect to varying filter sizes (1–10) and numbers of filter sizes (1–5), while keeping the total number of filters constant at 300 and filter sizes are ordered consecutively. Figure 3 shows that performance generally falls when increasing the filter size, and the best performance is achieved using three filters of sizes (2,3,4). Another important observation is that the variation in performance is not very substantial, implying that the model is fairly robust to the specific setting of this parameter.

4.2 Test results

	SVM		CNN	
Hallmark	**BoW**	**Rich**	**Base**	**Tuned**
Sustaining proliferative signaling	**70.0**%	67.4%	66.3%	67.9%
Evading growth suppressors	53.5%	65.3%	66.7%	**71.5**%
Resisting cell death	75.9%	82.7%	**86.9**%	86.7%
Enabling replicative immortality	73.1%	90.9%	91.2%	**91.5**%
Inducing angiogenesis	73.9%	**85.7**%	74.8%	79.4%
Activating invasion and metastasis	72.5%	72.7%	82.0%	**82.6**%
Genomic instability and mutation	71.2%	69.2%	72.2%	**81.7**%
Tumor promoting inflammation	69.9%	76.6%	81.6%	**84.2**%
Cellular energetics	78.1%	85.7%	76.6%	**88.3**%
Avoiding immune destruction	54.3%	71.8%	67.7%	**75.8**%
Average	69.2%	76.8%	76.6%	**81.0**%

Table 4: Comparison of test results using F-score

	SVM		CNN	
Hallmark	**BoW**	**Rich**	**Base**	**Tuned**
Sustaining proliferative signaling	88.6	88.9	**92.1**%	91.0%
Evading growth suppressors	87.9	91.7	94.8%	**96.4**%
Resisting cell death	92.4	95.5	97.1%	**97.7**%
Enabling replicative immortality	92.4	97.4	**99.8**%	99.5%
Inducing angiogenesis	94.7	98.4	97.9%	**99.1**%
Activating invasion and metastasis	96.0	94.0	97.8%	**98.2**%
Genomic instability and mutation	92.5	91.7	95.8%	**97.0**%
Tumor promoting inflammation	92.7	95.9	**98.3**%	98.1%
Cellular energetics	99.1	**99.6**	99.5%	**99.6**%
Avoiding immune destruction	94.6	96.1	97.8%	**99.1**%
Average	93.1	94.9	97.1%	**97.6**%

Table 5: Comparison of test results using AUC

The results of the evaluation on the test data are shown in Table 4 for F-score and 5 for AUC. Overall, both metrics agree that the SVM with bag-of-words features has the lowest performance, and the CNN tuned to the task the highest. As could be expected, the SVM with rich features outperforms the base CNN in terms of F-score; however, the latter, generic model achieves a notably higher AUC than the SVM, suggesting that the slight advantage of the former for F-score may be due in part to a better position of the decision boundary.

The CNN tuned to the task achieves the highest performance on average by both metrics, and further has the highest performance for $7/10$ individual classification tasks in terms of both F-score and AUC, outperforming the previous state-of-the-art on this dataset.

5 Discussion

Our evaluation contrasts methods separated by two methodological divides: discrete, interpretable, hand-engineered features vs. continuous, opaque, automatically learned features for one, and convex optimization vs. gradient descent in a complex landscape with many local minima for the other. The choice between the SVM representing the former choices and the CNN representing the latter is not necessarily only a question of which performs better, but also of methodological fit, both to the broader machine learning framework and for the practitioners applying the approach.

A key point of interest in neural methods is feature learning, i.e. their capacity to learn complex models with minimal manual effort in feature engineering. As shown again in our experiments, a CNN taking only document text and word embeddings induced from unlabeled text as input can outperform an SVM with extensive manually engineered features derived from sources such as syntactic analysis and named entity recognition. While the 3-4% point differences in AUC and F-score are positive results in favor of the CNN, the relative simplicity and generality of the model is arguably a greater advantage supporting the choice of the CNN over the feature-rich SVM — indeed, one might well argue that the

most interesting of our results is that the basic general CNN without any task or domain adaptation only narrowly loses to the SVM in F-score, and outperforms it in terms of AUC. The CNNs can be more readily adapted to other tasks and carry much fewer technical requirements: while the SVM system of Baker et al. (2016) requires running separate tools for lemmatization, parsing, and named entity tagging in addition to the machine learning method, the CNN has no such external dependencies.

For practitioners familiar with SVMs and domain NLP tools, it should be noted that the potential shift to neural methods is not without its own issues. As detailed by Zhang and Wallace (2015), even the simple CNN model considered here comes with a potentially overwhelming number of hyperparameters and related modeling and optimization choices, many of which have task-specific optima, and the cost of training and evaluating large numbers of model variants can be prohibitive even on modern GPU-based systems.[6] For machine learning researchers used to working with convex optimization problems, the random elements involved in training neural network models can also be a source of frustration, and the need to account for variance from network initialization and training also imposes additional computational costs.

Nevertheless, we believe that the simplicity, performance and rich potential for extension and further development of CNNs are more than sufficient to motivate further research on this class of models also in biomedical NLP and anticipate that many domain text classification tasks will see new state of the art results through the use of this class of neural networks.

6 Conclusions

In this study, we have considered the application of convolutional neural networks to the biomedical domain text classification task of identifying the hallmarks of cancer associated with publication abstracts.

Using a recently introduced corpus, we demonstrated that a CNN model taking only the document text and word representations induced from unannotated general-domain text as input can achieve competitive performance with a previously proposed SVM-based state-of-the-art classifier with rich manually engineered features including syntactic analyses and named entity recognition outputs. We further adapted the CNN to the task by oversampling positive examples to counteract the class bias, using word vectors induced on biomedical domain text, and optimizing the filter sizes through evaluation on the development set. The adapted model was shown to outperform the SVM, establishing a new state-of-the-art result for this dataset.

We make all of the resources involved in this study available under open source and open data licenses from `https://cambridgeltl.github.io/cancer-hallmark-cnn/`.

Acknowledgements

The first author is funded by the Commonwealth Scholarship and the Cambridge Trust. This work is supported by Medical Research Council grant MR/M013049/1 and the Google Faculty Award.

References

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical NLP. In *Proceedings of BioNLP*.

François Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

[6] We performed our CNN experiments on the Cambridge high performance computing cluster using NVIDIA K20 GPUs. Although individual epochs completed in 5 seconds on average and model training times were a few minutes at most, the repetitions and parameter grid involved training over 5000 models, and the total training time for the experiments was approximately 150 GPU-hours, not including preliminary and discarded experiments not reported here.

Mary E. Doland. 2014. Capturing cancer initiating events in OncoCL, a cancer cell ontology. In *AMIA Jt Summits Transl Sci*.

Rachel Ginn, Pranoti Pimpalkhute, Azadeh Nikfarjam, Apurv Patki, Karen OConnor, Abeed Sarker, Karen Smith, and Graciela Gonzalez. 2014. Mining twitter for adverse drug reaction mentions: a corpus and classification benchmark. In *Proceedings of BioTxtM 2014*.

Douglas Hanahan and Robert A Weinberg. 2000. The hallmarks of cancer. *cell*, 100(1):57–70.

Douglas Hanahan and Robert A Weinberg. 2011. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yann LeCun and Yoshua Bengio. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Hee-Jin Lee. 2014. Oncosearch: cancer gene search engine with literature evidence. *Nucl. Acids Res*.

Nut Limsopatham and Nigel Collier. 2016. Modelling the combination of generic and target domain embeddings in a convolutional neural network for sentence classification. In *Proceedings of BioNLP'16*, page 136.

Haibin Liu, Tom Christiansen, William A Baumgartner Jr, and Karin Verspoor. 2012. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *J. Biomedical Semantics*, 3:3.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013a. Distributional semantics resources for biomedical text processing. *Proceedings of LBM*.

Sampo Pyysalo, Tomoko Ohta, and Sophia Ananiadou. 2013b. Overview of the cancer genetics (CG) task of BioNLP Shared Task 2013. In *BioNLP Shared Task 2013 Workshop*.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of biomedical informatics*, 42(5):852–865.

Burr Settles. 2005. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of SemEval 2015*, pages 464–469.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Pontus Stenetorp, Hubert Soyer, Sampo Pyysalo, Sophia Ananiadou, and Takashi Chikayama. 2012. Size (and domain) matters: Evaluating semantic word space representations for biomedical text. In *Proceedings of SMBM'12*.

Lin Sun and Anna Korhonen. 2009. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 638–647. Association for Computational Linguistics.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.