Deep Belief Networks and Biomedical Text Categorisation

Antonio Jimeno Yepes[♢♠], **Andrew MacKinlay**[♢♠], **Justin Bedo**[♢♠], **Rahil Garnavi**[♢], **Qiang Chen**[♢]
[♢] IBM Research – Australia, 380 La Trobe Street, Melbourne, VIC, Australia
[♠] Dept. of Computing and Information Systems, University of Melbourne, Australia
{antonio.jimeno,admackin,justin.bedo,rahilgar,qiangchen}@au1.ibm.com

Abstract

We evaluate the use of Deep Belief Networks as classifiers in a text categorisation task (assigning category labels to documents) in the biomedical domain. Our preliminary results indicate that compared to Support Vector Machines, Deep Belief Networks are superior when a large set of training examples is available, showing an F-score increase of up to 5%. In addition, the training times for DBNs can be prohibitive. DBNs show promise for certain types of biomedical text categorisation.

1 Introduction

Text categorisation is the task of automatically assigning pre-defined labels to text. In the biomedical domain, research in automatic text categorisation has mostly taken place in the context of indexing MEDLINE® citations with Medical Subject Headings (MeSH®).

MEDLINE is the largest collection of biomedical abstracts and contains over 23M citations with over 800k new citations every year, making it difficult to keep up-to-date with new discoveries. To help cataloging and searching biomedical documents, the US National Library of Medicine (NLM)® has produced the MeSH controlled vocabulary with over 26k headings. At NLM, each MEDLINE citation is manually assigned a number of relevant medical subject headings enumerating the topics of the paper. Machine learning for text categorisation in this context is appealing due to the large number of citations available to train machine learning algorithms.

In text categorisation, the most frequently used feature representation is *bag-of-words*, where text is converted into a feature vector in which each dimension corresponds to a word or phrase and stores either a binary value indicating its presence in the document or a numerical value indicating its frequency (Apte et al., 1994; Dumais et al., 1998; Sebastiani, 2002). This relatively simple approach has proven to be robust enough (Jimeno-Yepes et al., 2011) that it is difficult to improve on its performance with more sophisticated representations. In this work, we explore the use of Deep Belief Networks (DBN) to automatically generate a new representation in biomedical text categorisation. Since DBNs have a richer internal representation than SVMs, we wished to evaluate whether this would lead to improved classification performance compared to *bag-of-words*.

2 Related work

There are several approaches being used for text categorisation in the biomedical domain trying to reproduce the manual MeSH indexing. NLM has developed the Medical Text Indexer (MTI) (Aronson et al., 2004; Mork et al., 2013), which is used to suggest MeSH headings for new citations to indexers. MTI combines MetaMap (Aronson and Lang, 2010) annotation and recommendations from similar citations recovered using the PubMed Related Citations (Lin and Wilbur, 2007) tool that are post-processed to comply with NLM indexing rules. Results for the most frequent categories, as used in this work, indicate that machine learning methods can produce better results than MTI (Jimeno Yepes et al., 2013). Recently, there has been interest in comparing MeSH indexing approaches in the BioASQ challenge.[1] It has been found that bag-of-word representations without feature selection already provide competitive performance.

Recently, several studies have utilised different deep learning methods to build multiple layers of feature representation for documents, such as a Stacked De-noising Autoencoder (SDA) (Vincent et al., 2010; Glorot et al., 2011) and a DBN (Hinton

[1] http://www.bioasq.org/workshop/schedule

Antonio Jimeno Yepes, Andrew MacKinlay, Justin Bedo, Rahil Garvani and Qiang Chen. 2014. Deep Belief Networks and Biomedical Text Categorisation. In *Proceedings of Australasian Language Technology Association Workshop*, pages 123−127.

and Salakhutdinov, 2006) for tasks including spam filtering (Tzortzis and Likas, 2007). In this work, we apply DBN as our deep learning algorithm for biomedical text categorisation, trying to reproduce MeSH indexing for the 10 top most frequent MeSH headings.

3 Methods

3.1 Deep Belief Networks

Restricted Boltzmann Machines (RBM) A (restricted) Boltzmann Machine (RBM) (Salakhutdinov et al., 2007) is a parameterised generative model representing a joint probability distribution. Given some training data, learning an RBM means adjusting the RBM parameters to maximise the likelihood of the training data under the model. Restricted Boltzmann machines consist of two layers containing visible and hidden neurons.

The energy function $E(v,h)$ of an RBM is:

$$E(v,h) = -b'v - c'h - h'Wv; \qquad (1)$$

where W represents the weights connecting hidden and visible units and b, c are the offsets of the visible and hidden layers respectively. The joint probability distribution is then given by the exponential family $P(v,h) = \frac{1}{Z}e^{E(v,h)}$, where Z is a normalisation factor. The likelihood of some data $X \subset \mathbb{R}^n$ is thus $\mathscr{L}(X) := \prod_{v \in X} \sum_h P(v,h)$ and b, c, and W are chosen to maximise this likelihood (or equivalently minimise the negative log likelihood):

$$\arg_{b,c,W} \min -\log \mathscr{L}(X) = -\sum_{v \in X} \log \sum_h P(v,h).$$

We used the Contrastive Divergence method (Hinton, 2002) to find an approximate solution.

Deep Belief Network A DBN normally is the stack of many layers of RBM model. Hinton and Salakhutdinov (2006) showed that RBMs can be stacked and trained in a greedy manner to form so-called Deep Belief Networks (DBN). DBNs are graphical models which learn to extract a deep hierarchical representation of the training data.

The hidden neurons extract relevant features from the observations, and these features can serve as input to another RBM. By stacking RBMs in this way, we can learn a higher level representation.

Practical training strategies In practice, the DBN training often consists of two steps: greedy layer-wise pretraining and fine tuning. Layer-wise pretraining involves training the model parameters layer by layer via unsupervised training. Fine tuning is achieved by supervised gradient descent of the negative log-likelihood cost function.

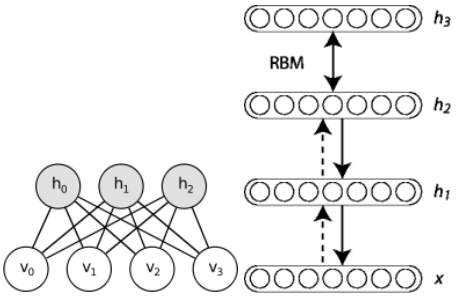

Figure 1: Deep Neural Network (left) and RBM (right)

The DBN implementation used in this work has been obtained from `http://www.deeplearning.net/tutorial` built on Theano[2]. Text data is very sparse with only a few dimensions having non-zero values. We modified the DBN code to deal with sparse matrices.

3.2 Support Vector Machine

We used a Support Vector Machine (SVM) with a linear kernel as our baseline method. SVM has shown good performance on text categorisation (Joachims, 1998) as well as in MeSH indexing (Jimeno Yepes et al., 2013) and within BioASQ. In this work, we have used the implementation from the MTI ML package[3] that follows the work of (Zhang, 2004) and uses Hinge loss with stochastic gradient descent.

3.3 Data set

Training and test sets have been obtained from the MTI ML site. There are 24,727 training citations and 12,363 test citations. From these data sets, we have selected the top 10 most frequent MeSH headings available from Table 1.

We have also used a larger set since we realised in the earlier stages of experimentation that more data was needed to train the DBN. This larger set has been obtained from the NLM Indexing Initiative[4] and is split into 94,942 training citations and 48,911 test citations. Results on both sets are reported for the same categories.

We processed the citations to extract the text from the title and the abstract. From the text, we

[2] `http://deeplearning.net/software/theano`
[3] http://ii.nlm.nih.gov/MTI_ML
[4] http://ii.nlm.nih.gov/DataSets/index.shtml#2013_MTI_ML

extracted tokens using a regular expression looking for white spaces and punctuation marks. Tokens were lowercased and filtered using a standard stopword list. Binary values for the features (present or absent) are considered. Only tokens that appear in at least two citations in the training set were considered, considerably reducing the number of features.

4 Results

The SVM and the DBN were trained and tested on the data sets. Binary classifiers predicting each individual category were trained for each one of the selected MeSH headings. For DBN, we used 2/3 of the training data for unsupervised pretraining and 1/3 for fine tuning the model due to DBN training cost, while for SVM we used all the training data.

Configuring the DBN requires specifying the number of hidden layers and the number of units per layer. We used one hidden layer to give three layers in total. We used two different configuration of training units, set empirically (and semi-arbitrarily) from data samples – *DBN 250* with 250 units in each of the three layers and *DBN 500*, with 500 units per layer.

Tables 1 and 2 show results for the small set with 16000 for DBN pretraining and 8727 for fine tuning and the large set with 63294 for DBN pretraining and 31647 for fine tuning.

As shown in Table 1, with the smaller datasets, SVM performance is superior to DBN, however DBN substantially outperforms SVM on the six most frequent categories. DBN results are much lower when the categories are less frequent and for *Adolescent*, DBN simply classified all citations as negative. *DBN 500* performs better than *DBN 250* in the top six most frequent categories.

Figure 2 shows learning curves obtained by training the three methods on increasingly large subsets of the small training set. SVM outperforms DBN when there is limited training data, but as the amount of training data is increased, for certain categories DBN, especially *DBN 500*, surpasses SVM (as expected from Table 1).

Results were obtained using the same subset and it could be interesting to see the behavior if different subsets of the training data are used. DBN results depend as well on the partition of the training data, using all the data for pretraining and fine tuning the perfomance of DBN on the small set improves (avg. F1: 0.7282).

Table 2 shows that when there is more training data available, the performance penalty for the DBN methods versus SVM over the sparser categories disappears. In addition, there is also less of a difference between results of 250 and 500 units per layer. Overall all three methods are more similar to one another over this larger data set, with better results for DBN on average. Absolute results between Tables 1 and 2 are not comparable since two different collections are used, e.g. some categories have significantly different performance.

5 Discussion

In our experiments, DBN overall performance is comparable to SVM with a linear kernel being better in some of the categories when a large set of training data is used. We also evaluated SVM with Radial Basis Function kernel (not reported) but the results were comparable to a linear kernel.

Compared to image processing, text categorisation has a larger dimensionality that varies with the size of the data set since there is the chance of finding new unique words, even though data is sparse and few of the citation features have a value. On the small set, with a batch size of 200 citations, the number of unique features is 2,531 and with a batch size of 8,000 it is 26,491, while in the larger set, 53,784 unique features were found.

6 Conclusions and Future Work

DBN shows competitive performance compared to SVM. We have tried a limited set of configurations with only one hidden layer. Deeper configurations with a more varied number of units can be explored but the pretraining phase is expensive. We would like to explore different pretraining and supervised tuning ratios to reduce training time. In addition, identifying the best DBN configuration can be expensive. (Rahimi and Recht, 2009) suggest approaches to avoid an explosion of possibilities which could be useful here.

Deep learning requires a significant amount of time to train, e.g. SVM was trained in several minutes while the DBN pretraining in the large set took five days. To alleviate this, we could consider methods to reduce dimensionality (Weinberger et al., 2009; Bingham and Mannila, 2001). Nonetheless, we believe that this work shows that DBNs show promise for text categorisation, as they are able to provide superior performance to SVM-based techniques traditionally for such tasks.

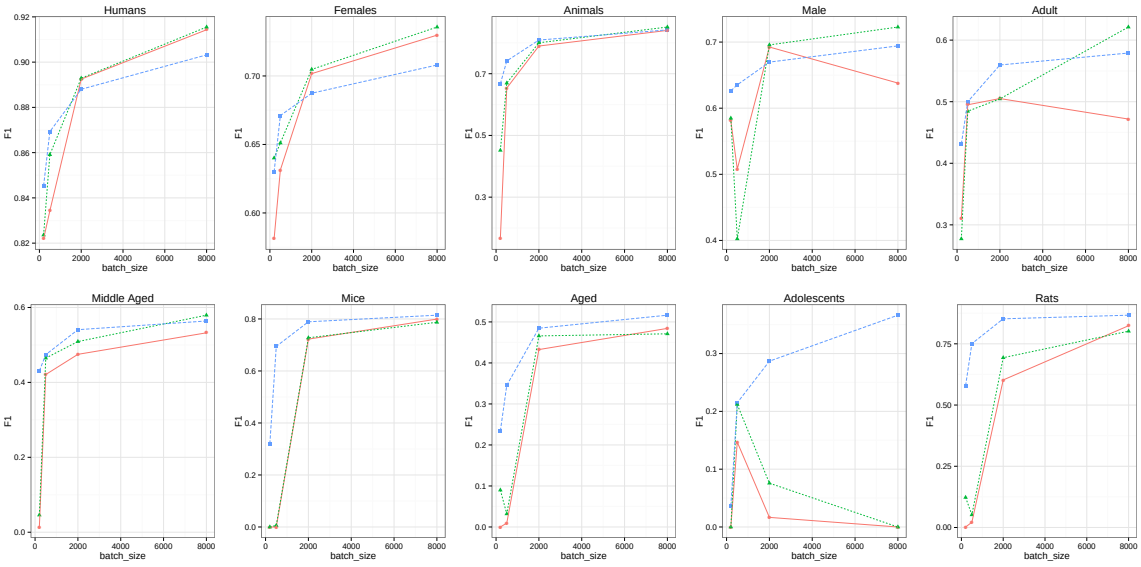

Figure 2: Training size vs F1 on the small set. There is one plot per category. Three methods are shown: SVM (slashed blue line, square shaped point), DBN with three layers with 250 units each (continuous red line, round shaped point) and DBN with three layers with 500 units each (dotted green line, triangle shaped point).

	Methods	SVM	(linear)		DBN	250		DBN	500	
Category	Positives	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Humans	7688	0.8983	0.9083	0.9032	0.9016	0.9273	0.9143	**0.9032**	**0.9282**	**0.9155**
Female	4616	**0.7215**	0.6950	0.7080	0.7001	0.7621	0.7298	0.6945	**0.7821**	**0.7357**
Male	4396	0.7034	0.6852	0.6942	0.4771	0.9627	0.6380	**0.7138**	**0.7318**	**0.7227**
Animals	4347	0.8585	0.8261	0.8420	**0.8797**	0.8042	0.8403	0.8476	**0.8548**	**0.8512**
Adult	2518	0.6092	0.5516	0.5790	**0.6397**	0.3737	0.4718	0.6098	**0.6330**	**0.6212**
Middle Aged	2108	0.5978	**0.5337**	0.5639	**0.7108**	0.4255	0.5323	0.7085	0.4900	**0.5794**
Aged	1467	0.5684	**0.4731**	**0.5164**	0.6806	0.3758	0.4842	**0.6813**	0.3599	0.4710
Mice	1304	0.8588	**0.7745**	**0.8145**	0.8102	0.7891	0.7995	**0.8890**	0.7063	0.7872
Adolescent	1066	**0.4059**	**0.3340**	**0.3664**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Rats	938	**0.9118**	**0.8262**	**0.8669**	0.8633	0.7878	0.8239	0.8702	0.7431	0.8016
Average	3045	**0.7134**	**0.6608**	**0.6861**	0.6663	0.6208	0.6428	0.6918	0.6229	0.6556

Table 1: Text categorisation results for the 10 selected categories with the small set and a batch size of 8000 citations. Results are reported in precision (Pre), recall (Rec) and F-measure (F1). Average results are shown at the bottom of the table. *DBN 250* means using three layers with 250 units each. *DBN 500* means using three layers with 500 units each.

	Methods	SVM	(linear)		DBN	250		DBN	500	
Category	Positives	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Humans	35967	0.9052	0.9354	0.9201	**0.9209**	0.9436	0.9321	0.9204	**0.9445**	**0.9323**
Female	16483	0.7464	**0.7176**	0.7317	**0.8305**	0.6964	0.7576	0.8216	0.7160	**0.7652**
Male	15530	0.7267	0.6889	0.7073	**0.7917**	0.7025	0.7444	0.7878	**0.7213**	**0.7531**
Animals	11259	0.8431	**0.7613**	**0.8001**	0.8895	0.6879	0.7758	**0.9407**	0.6337	0.7573
Adult	8792	0.5824	**0.5296**	**0.5547**	**0.6915**	0.4480	0.5438	0.6696	0.3592	0.4676
Middle Aged	8392	0.6323	0.5728	0.6011	0.7239	0.5654	0.6349	**0.7375**	**0.5853**	**0.6527**
Aged	6151	0.5616	**0.5079**	**0.5334**	**0.7147**	0.4076	0.5191	0.6937	0.4303	0.5312
Adolescent	3824	0.4641	**0.3690**	**0.4111**	0.5735	0.2529	0.3510	**0.6583**	0.2202	0.3300
Mice	3723	0.8386	0.7284	0.7796	0.8746	0.7268	0.7939	**0.8984**	**0.7295**	**0.8052**
Rats	1613	0.8461	**0.7601**	0.8008	**0.9150**	0.7204	0.8061	0.9123	0.7421	**0.8185**
Average	11173	0.7146	**0.6571**	0.6847	0.7926	0.6152	**0.6927**	**0.8040**	0.6082	0.6926

Table 2: Text categorisation results for the 10 selected categories with the large set and a batch size of 31647 citations. Results are reported in precision (Pre), recall (Rec) and F-measure (F1). Average results are shown at the bottom of the table. *DBN 250* means using three layers with 250 units each. *DBN 500* means using three layers with 500 units each.

References

Chidanand Apte, Fred Damerau, and Sholom M Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12:233–251.

Alan R Aronson and François-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Alan R Aronson, James G Mork, Clifford W Gay, Susanne M Humphrey, and Willie J Rogers. 2004. The NLM indexing initiative's medical text indexer. *Medinfo*, 11(Pt 1):268–72.

Ella Bingham and Heikki Mannila. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM.

Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

GE Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 1800:1771–1800.

Antonio Jimeno-Yepes, Bartłomiej Wilkowski, James G Mork, Elizabeth Van Lenten, Dina Demner Fushman, and Alan R Aronson. 2011. A bottom-up approach to MEDLINE indexing recommendations. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1583. American Medical Informatics Association.

Antonio Jose Jimeno Yepes, James G Mork, Dina Demner-Fushman, and Alan R Aronson. 2013. Comparison and combination of several MeSH indexing approaches. In *AMIA Annual Symposium Proceedings*, volume 2013, page 709. American Medical Informatics Association.

Thorsten Joachims. 1998. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK. Springer-Verlag.

Jimmy Lin and W John Wilbur. 2007. PubMed related articles: a probabilistic topic-based model for content similarity. *BMC bioinformatics*, 8(1):423.

James G Mork, Antonio Jimeno-Yepes, and Alan R Aronson. 2013. The NLM Medical Text Indexer system for indexing biomedical literature. In *BioASQ@ CLEF*.

Ali Rahimi and Benjamin Recht. 2009. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Grigorios Tzortzis and Aristidis Likas. 2007. Deep belief networks for spam filtering. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 2, pages 306–309. IEEE.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM.

Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 116–, New York, NY, USA. ACM.