CNNs for Sentence Classification Applied to Literary Works

Yves Blain-Montesano and **Michaela Socolof**
McGill University
1085 Dr. Penfield Avenue
Montréal, QC, Canada H3A 1A7
yves.blain-montesano@mail.mcgill.ca
michaela.socolof@mail.mcgill.ca

1 Abstract

We implement a convolutional neural network (as described in (Kim, 2014)) to the task of sentence subjectivity classification, and we compare its performance on non-fiction and fiction data sets. In doing so, we present a new annotated data set containing the first 1,000 sentences of Jane Austen's *Sense and Sensibility* labeled for subjectivity. We find that the model does not generalize well to the fiction case, which we speculate may be due to frequent perspective shifts in fiction.

2 Introduction

Kim (2014) implements a convolutional neural network and applies it to various sentence classification tasks. They demonstrate that CNNs are capable of achieving good performance on these tasks and that word embeddings trained on unrelated tasks nevertheless hold useful information. We examine the performance of such a model on literary (fiction) data, looking specifically at the task of classifying sentences as subjective or objective. Sentence classification for subjectivity in fiction can be useful for tasks such as narrative summarization, e.g. generating plot summaries, as well as information extraction and sentiment analysis. We hypothesized that word embeddings trained on non-fiction data would not generalize well to the fiction case, where there is a wide range of stylistic variation, particularly in terms of narrative perspective.

We use the non-fiction Subjectivity[1] data set (Pang and Lee, 2004) of 10,000 sentences and a hand-labeled data set consisting of 699 sentences of Jane Austen's *Sense and Sensibility*[2]. We first evaluate performance on non-fiction, then examine how fiction data can affect performance on this task. For each genre, we evaluate both a CNN with randomly initialized (but trained) embeddings, and one using pretrained GloVe[3] (Pennington et al., 2014) embeddings.

A convolution kernel over words might be seen as analogous, at least partially, to an n-gram model. Instead, however, we learn larger word context features via convolution (Collobert et al., 2011). Furthermore, by using word representations that contain relatedness information that is indicated by proximity in a vector space, this may allow learning kernels that distinguish semantic features.

3 Related work

CNN models have been used to great success in a variety of NLP tasks, from semantic parsing to sentence modelling (Yih et al., 2014; Shen et al., 2014; Kalchbrenner et al., 2014; Collobert et al., 2011). In (Kim, 2014), as described above, CNNs are applied to a variety of sentence classification tasks. The purpose of the current work is to investigate whether these methods are as effective on a different genre of text, namely fiction.

Applying NLP to fiction data sets has gained traction in recent years, and there is a steadily increasing amount of work intended to facilitate this process.

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[2] https://github.com/yblainm/comp550project

[3] https://nlp.stanford.edu/projects/glove/

For instance, BookNLP[4] (Bamman et al., 2014) is an NLP pipeline specifically created to scale to books and other long works, and has been used to perform such tasks as inferring character types in novels. We look specifically at subjectivity classification, which can be seen as a type of sentiment analysis, and which has been explored in detail in (Pang and Lee, 2004), in order to see whether current methods generalize well to fiction.

4 Model

The architecture of the model we and (Kim, 2014) use borrows techniques from (Collobert et al., 2011), such as its use of max-over-time pooling. The model takes as input a matrix of word embeddings where each row is a word vector of dimensionality k. The sentence matrix is padded to length l, that of the longest sentence seen in all input data[5].

A sentence is therefore a $l \times k$ matrix where $x_i \in \mathbb{R}^k$ is the word embedding vector for the i-th word in the sentence.

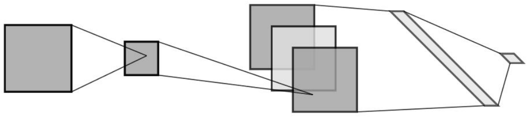

Figure 1: Single-channel CNN-non-static architecture from (Kim, 2014). The first layer represents a sentence's encoded word tokens. The second layer provides a matrix of word embeddings for a sentence to the third layer of parallel convolutional layers, whose outputs are pooled and then fed to class output via a dense layer.

Multiple convolutional layers (using ReLU activation) are applied to the sentence matrix in parallel (as in the third layer of fig. 1), each with different filter heights, i.e. $h_i \in \mathbb{N}^+$ for the i-th parallel layer, and each filter is a $w \in \mathbb{R}^{h_i \times l}$. See section 5, fig. 5 for our chosen filters and other hyperparameters. We use a stride of 1.

Each feature map is pooled as per (Collobert et al., 2011) to retrieve the most activated kernel's value for each feature map (e.g., if there are 5 different filter heights and 10 filters each, there will be a 50-dimensional vector as a result of pooling).

This is used as input for a dense layer using softmax activation with n output neurons for n classes to predict. Dropout is applied, as is a constraint on the maximum L2 norm for the dense layer's weights.

The model is trained using the AdaDelta SGD optimizer. We considered using Adam but wished to compare our results to those they achieved on the subjectivity data set. We manually found 50 as an optimal batch size, as used in (Kim, 2014).

5 Methodology

We implement the above-described model in Keras (Chollet and others, 2015). Note that in our implementation, vocabulary of both training and test data is known in advance—this was not specified by Kim (2014)—and that a word embedding for the padding symbol is learned. These both represent weaknesses of our implementation.

We perform a hyperparameter search on the subjectivity data set, of which 10% is reserved for validation and testing respectively. We use a batch size of 256 for the search. We include in the search space the best hyperparameters found by Kim (2014), though we search over only a subset of possible hyperparameters for computational reasons. Equally, we search over varying filters per convolutional layer, to potentially minimize overfitting if a smaller number of filters is desirable.

Max L2 norm	0.5, 3, 10
Filter heights	(1,2,3), (2,4,6), (1,4,8), (1,2,3,4,5), (3,6,10)
Num. filters	20, 50, 100, 200
Dropout Prob.	0.25, 0.5, 0.9

Figure 2: Hyperparameter range

Of note is our search over varying filter heights. Assuming we learn meaningful word embeddings (or inherit them from GloVe), there may be multiple related words whose vectors will activate in the same context around a word. That is, semantically, these words may act similarly in similar contexts. Assuming this is learned, it may be beneficial to learn larger

[4] https://github.com/dbamman/book-nlp

[5] This works for our implementation, but would require one to either preempt sentence length, which excludes some valid written language of arbitrary length, or avoid use cases where input data will be unknown in advance. Fully convolutional networks (Long et al., 2015) can avoid the problem of fixed input dimensions, however.

	Train	Val.	Test
Subj	8000	1000	1000
Austen	559	N/A	140
Mixed	2559	N/A	140

Figure 3: Data set samples

Subj-Subj	50
Subj-Austen	50
Austen-Austen	1
Mixed-Austen	50

Figure 4: Training batch size per experiment. The first word is the data set trained on, and the second is which was tested on.

features, as words' similar representations could reduce the effective noisiness of these large windows.

We evaluate the model in the following ways: trained on the subjectivity data set, and tested on its test set; trained on all of the subjectivity set, and tested on all of our hand-labeled data; trained on a split of our hand-labeled data, and tested on its test split; and finally trained on a mixture (2,000 and 559 samples from Subj and Austen respectively) and tested on 140 samples our hand-labeled data.

We hand-labeled the Austen data set by following, to the best of our understanding, the method described in (Pang and Lee, 2004). However, as Riloff and Wiebe (2003) state, "It is [very hard] to obtain collections of individual sentences that can be easily identified as subjective or objective." It is therefore likely that obtaining annotations for the same data set would yield a relatively high degree of variability.

The choice of a batch size of 1 for the Austen-Austen task (effectively online stochastic gradient descent) was primarily due to the very small amount of data when training. As the data is sparse and the loss curve likely not convex, the noisiness of data and variability of the computed gradient provided by online SGD may allow us to escape local minima and saddle points (Ge et al., 2015).

For each task, we evaluate the model using randomly initialized embeddings in the range of $[-0.05, 0.05]$, and using the pretrained GloVe embeddings trained on Wikipedia 2014 and Gigaword 5. These were picked rather than those trained on Common Crawl[6], as we expect Wikipedia data and the news data used for Gigaword to be mostly objective, non-fiction data. Given that we attempt to examine whether subjectivity can be learned similarly in fiction and non-fiction, this was found to be necessary.

6 Results

As part of our work, we have created a small hand-labeled data set of the first 1,000 sentences of Jane Austen's *Sense and Sensibility*. Of these sentences, 47.7% were labeled as objective and 52.3% as subjective.

Our hyperparameter search appears to indicate that learning larger filter kernels is beneficial for this task, at least on the subjectivity data set, and that a larger number of filters than found in (Kim, 2014) is not detrimental. That a dropout probability of 0.25 (Kim (2014) obtains 0.5) is found indicates that perhaps regularization need not be as powerful, though the small size of our data should make our model prone to overfitting.

Max L2 norm	3
Filter heights	(3,6,10)
Num. filters	200
Dropout Prob.	0.25

Figure 5: Chosen hyperparameters

We obtain similar results (see fig. 6 for a table of results) to Kim (2014) on the subjectivity task, though our baseline (with random initial embeddings) performs non-negligeably better than theirs (0.919 to their 89.6). Inclusion of pretrained word vectors from GloVe improves performance to a level similar to their word2vec-initialized model (0.931 to their 0.934).

The model performs slightly above chance level on the Subj-Austen task, at a relatively good performance for the Austen-Austen task (approx. 0.75), and slightly noticeably worse on the Mixed-Austen task (approx. 0.72).

7 Discussion

We find that subjectivity features learned from non-fiction data do not generalize well to fiction. It may be that our model learns surface cues that are more prominent in modern non-fiction. It is not clear to us

[6] http://commoncrawl.org/

CNN-rand (rand. embed.)			0.896
CNN-non-static (word2vec)			0.934
Task	GloVe	Best acc./ep.	Final acc./ep.
S-S	no	N/A	0.919 / 15
S-S	yes	0.931 / 10	0.927 / 20
S-A	no	0.598 / 5	0.584 / 50
S-A	yes	0.584 / 9	0.583 / 50
A-A	no	0.75 / 12	0.714 / 25
A-A	yes	0.736 / 13	0.707 / 25
M-A	no	N/A	0.721 / 25
M-A	yes	0.643 / 11	0.607 / 25

Figure 6: Accuracy per task, with and without GloVe. Subj, Austen, and Mixed abbreviated to S, A and M. Best acc. is the accuracy and epoch which we consider had best performance. Results from (Kim, 2014) on the subjectivity data set are added for comparison to the first task. Final acc. is the accuracy reached after all epochs of training. Best acc. is N/A if the final one is greater.

whether it is an effect of a particular author's style, an effect of language evolution, or an effect of some property inherent to narrative works of fiction.

Of note is that features learned even from our small amount of hand-labeled data on the Austen-Austen task generalize well to data of the same work and author.

Mixed data decreases performance noticeably, though not remarkably. It seems to indicate, however, that additional non-fiction data does not add useful information for our particular model. We informally hypothesize that our model is learning surface information which is particular to a given style of writing, and that a mixture of data only confuses this.

In general we find the use of non-fiction pretrained word vectors from GloVe (Wikipedia 2014 and Gigaword 5) to harm performance on literary data. In following with our above hypothesis, co-occurrence information provided by GloVe might be ”informing away” some underlying semantic or surface information of certain words, thereby hampering perfomance. The fact that performance is slightly increased by GloVe on the Subj-Subj task indicates that non-fiction information contained in it aids in non-fiction classification in particular (though not very noticeably).

8 Conclusion

We find that classification on the subjectivity data set (Pang and Lee, 2004) on fiction is possible even with little data using a CNN architecture from (Kim, 2014). Using word-embeddings pretrained on non-fiction aids non-fiction classification, but either hampers or achieves nothing on fiction classification when the model is trained on non-fiction. Mixing non-fiction and fiction training data decreases performance, especially when using pretrained non-fiction word embeddings.

We recommend further inquiry into the effect of authorial style and dialectal/language-evolutionary differences in subjectivity classification. For example, it may be the case that the model described in this paper would do better with fiction texts written in a more distant third-person point of view, where subjective statements might be more clearly flagged, than first-person or close third-person texts. Furthermore, to allow learning embeddings and features of semantics rather than (possibly) only surface features, we recommend the creation of more labeled literary data. Also note that we make our hand-labeled data as well as code available on a public repository.

Statement of Contributions

Both authors were involved in the project design and the interpretation of results, and both contributed to writing the report. As for the experiments themselves, M. S. hand-annotated the fiction data set, and Y. B.-M. wrote the model implementation and experiment code.

Appendix

This appendix contains loss and accuracy curves for evaluated tasks. It does not contain the Subj-Subj tasks, however.

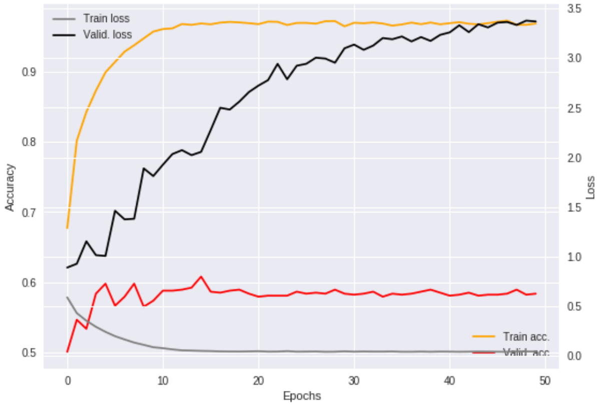

Figure 7: Subj-Aust, no GloVe

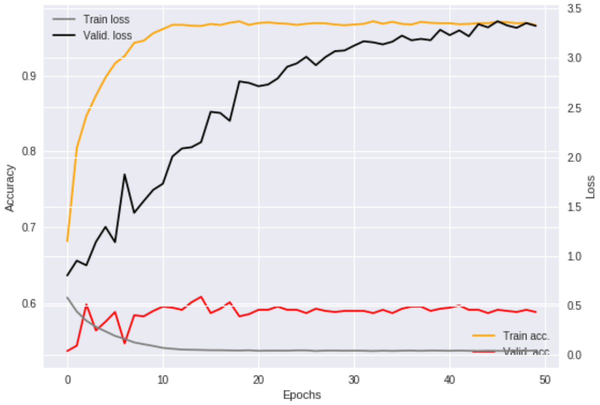

Figure 8: Subj-Aust with GloVe

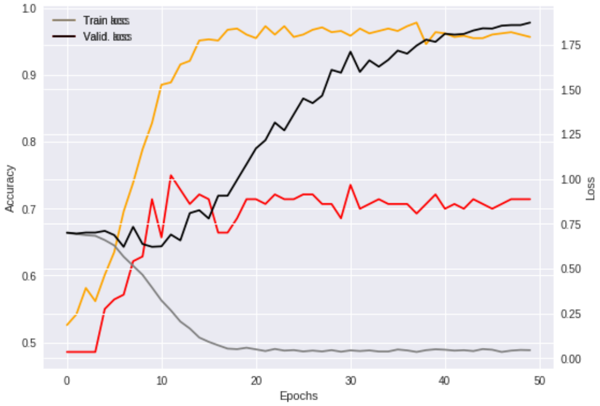

Figure 9: Aust-Aust, no GloVe

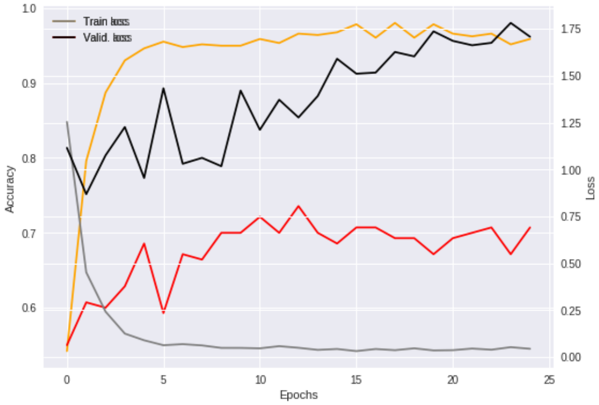

Figure 10: Aust-Aust with GloVe

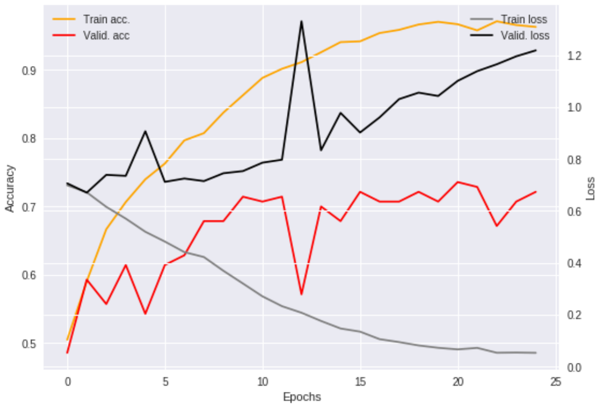

Figure 11: Mixed-Aust, no GloVe

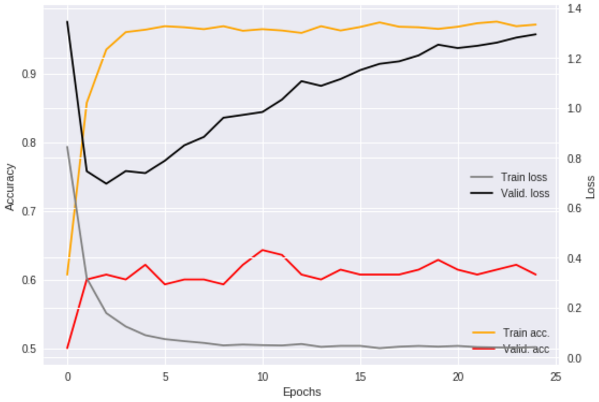

Figure 12: Mixed-Aust with GloVe

References

David Bamman, Ted Underwood, and Noah Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of ACL 2014*.

François Chollet et al. 2015. Keras. https://keras.io.

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.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. 2015. Escaping from saddle points — online stochastic gradient for tensor decomposition. In Peter Grnwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 797–842, Paris, France, 03–06 Jul. PMLR.

N. Kalchbrenner, P. Grefenstette, and P. Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL 2014*.

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

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP*.

Y. Shen, J. He, L. Gao, and G. Mesnil Deng. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of WWW 2014*.

W. Yih, X. He, and C. Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of ACL 2014*.