A Calibration Method for the Evaluation of Sentiment Analysis

F. Sharmila Satthar, Roger Evans and Gulden Uchyigit
Computing, Engineering and Mathematics
University of Brighton
Brighton, UK
{F.Satthar,R.P.Evans,G.Uchyigit}@brighton.ac.uk

Abstract

Sentiment analysis is the computational task of extracting sentiment from a text document – for example whether it expresses a positive, negative or neutral opinion. Various approaches have been introduced in recent years, using a range of different techniques to extract sentiment information from a document. Measuring these methods against a gold standard dataset is a useful way to evaluate such systems. However, different sentiment analysis techniques represent sentiment values in different ways, such as discrete categorical classes or continuous numerical sentiment scores. This creates a challenge for evaluating and comparing such systems; in particular assessing numerical scores against datasets that use fixed classes is difficult, because the numerical outputs have to be mapped onto the ordered classes. This paper proposes a novel calibration technique that uses precision vs. recall curves to set class thresholds to optimize a continuous sentiment analyser's performance against a discrete gold standard dataset. In experiments mapping a continuous score onto a three-class classification of movie reviews, we show that calibration results in a substantial increase in f-score when compared to a non-calibrated mapping.

1 Introduction

Sentiment analysis is the computational study of people's opinions, appraisals, emotional attitudes toward entities, events and their attributes. The sentiment analysis task involves classifying texts according to the sentiment content they contain. Sentiment analysis is a very active research area in natural language processing, with many research projects working on building sentiment classifiers using different techniques and algorithms. Evaluation is an important process when estimating the performance of text/data classification in information retrieval or natural language processing systems. The accuracy of a (binary) classifier is typically measured based on its *precision*, *recall* and *f-score* values when applied to a gold standard dataset. This approach has been adopted for the evaluation of sentiment analysis systems too (Turney, 2002; Pang et al., 2002; Nasukawa and Yi, 2003; Prabowo and Thelwall, 2009), but it is complicated by the fact that sentiment analysis is usually a multi-class classification task.

Many sentiment analysis approaches focus on three classes such as *positive*, *negative* and *neutral*. However Saif et al. (2016) introduced an extra class, in addition to the neutral class, called *mixed-sentiment*, which is a mixture of positive and negative opinions, while Pang and Lee (2005) and Nakov et al. (2016) explored 4 or 5 star scales/classifications. To evaluate these types of multi-class classification tasks, precision, recall and f-score values are calculated for each class separately, and the performance measures for the whole system are then calculated by averaging those values using micro or macro-averaging (Prabowo and Thelwall, 2009).

Most supervised machine learning methods for sentiment analysis produce categorical outputs such as *positive*, *negative* and *neutral*, with no assumptions about the relationship between classes; they simply map texts into classes by associating text features with class labels. But other multi-class systems use rated or scaled methods so that their categorical outputs are implicitly ordered in a natural 'sentiment order' based on sentiment polarity and/or magnitude/intensity, such as the fol-

Proceedings of Recent Advances in Natural Language Processing, pages 652–660,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_084

lowing examples:

Positive > Neutral > Negative

Strong-Positive > Positive > Weak-Positive > Neutral>
Weak-Negative > Negative > Strong-Negative

3 stars > 2 stars > 1 star

In addition, some sentiment analysis applications are based more explicitly on sentiment scores, rather than sentiment classes, and produce numerical values with positive and negative signs as the output for a given text, such as $+0.987$, -0.786 ... or $+187$, -243 ... etc. Such methods typically use the sign to indicate the polarity of the given text and numerical values to define the sentiment strength (generally over a system-dependent range), with a sentiment value of 0 indicating a neutral text. A simple mapping from such scores to a 3-class sentiment model just uses the sign (+,0,-) to identify sentiment classes (positive, neutral, negative). However, there is no correspondingly simple way to use the magnitude to extend this to more classes (such as *'strong positive'*, *'weak positive'*, *'positive'* ... etc.), and no clear justification for the implicit claim that *neutral* is a single point (0). This paper introduces a method to address these concerns, by calibrating the mapping from a numerical score to a semantic class in a way that optimises the system's performace as a multi-class classifier.

To transform a numeric scale to an ordinal (categorical) scale, boundaries (upper and lower) for each sentiment class needed to be identified from the given numeric scale. These boundary values are 'cut-off values' for the sentiment classes, and are the parameters for a multi-class sentiment classification system based on the numerical scores. This paper proposes new techniques to assign cut-off values for each class using a learning-based evaluation technique. This transformation allows us to both optimise and evaluate a system that gives numeric outputs against a gold standard dataset that contains fixed categorical outputs.

We use evaluation performance measures (precision and recall) on a training subset of the dataset to adjust the parameters to produce an optimal result, by using Precision vs Recall (PR) curve visualisation. The parameters are optimised to give the best performance on the training set, and then evaluated using test set. In addition we can determine how far misclassified texts deviate from actual classes in multi-class ordered classification tasks, by computing macro-averaged mean absolute error which is the popular approach for ordinal classification (Nakov et al., 2016; Baccianella et al., 2009; Gaudette and Japkowicz, 2009).

We demonstrate our technique for tuning the parameters using the *Galadriel* sentiment analysis system (Satthar, 2015), which we built for sentiment analysis using an inheritance-based lexicon. *Galadriel* is an example of a class of systems which calculate sentiment scores by combining raw lexical scores using a range of arithmetic rules (summming, scaling, averaging etc.). The final output of *Galadriel* for a text is a signed real number which reflects sentiments expressed by the lexical items in quite a complex way, making the intepretation of scores as classes challenging. The calibration method achieves this mapping in an optimal way.

In this paper, section 2 discusses relevant previous research, in particular pre-evaluation processes and some general methods involved in sentiment classification (section 2.1) and use of the PR curve for evaluation (rsection 2.2). In section 3, we present our novel techniques for tuning the parameters. In section 4, we present our experiments with the *Galadriel* system, and the results of optimising cut-off values for sentiment classes. Section 5 compares the evaluation results using the cut-off values which are computed in the previous section with evaluation without calibration. Finally, section 6 provides the conclusion.

2 Related Work

2.1 Approaches to sentiment classification

Sentiment classification is most simply expressed as a two-class (*positive* and *negative*) or three-class (including *neutral*) classification problem. In recent work, sentiment analysis researchers have also been interested in greater than three class classification such as *strong positive* to *strong negative* and *scale-1* to *scale-5* (Aly, 2005; Lee and Grafe, 2010; Pang and Lee, 2005).

For supervised machine learning methods, the classes come directly from the labelled training data, which means that such systems can directly produce *positive* or *negative* labelled outputs without any direct intepretation of what the classes 'mean' (Pang et al., 2002; Hsu et al., 2010). Similarly, unsupervised learning methods directly produce *positive* or *negative* labelled outputs using different techniques and algorithms such as k-

Word	Bing Liu	Harvard GI	Vader	SentiWordNet	SenticNet	Taboada
good	+1	POS	+1.9	0.75 (POS)	+0.883	+3
glad	+1	POS	+2.0	0.5 (POS)	+0.413	+2
incapable	−1	NEG	−1.6	0.625 (NEG)	−0.736	−1
sad	−1	NEG	−2.1	0.25(NEG)	−0.306	−2
bad	−1	NEG	−2.5	0.875(NEG)	−0.367	−3

Table 1: Some lexical entries with their semantic orientation according to different lexicon dictionaries.

means, TF-IDF and PMI-IR algorithms (Turney, 2002; Zagibalov and Carroll, 2008; Unnisa et al., 2016), but again without a clear interpretation of the classes identified. Lexicon-based approaches (as well as some unsupervised learning methods methods such as Turney (2002)) have proceeded by calculating the semantic orientation (a numerical score) and deciding the polarity of the document depending on its sign and the sentiment strength based on its magnitude. Such methods calculate the semantic orientation of a document by the aggregating semantic orientation of words or phrases, using various arithmetic combinations of scores (Taboada et al., 2011; Palanisamy et al., 2013).

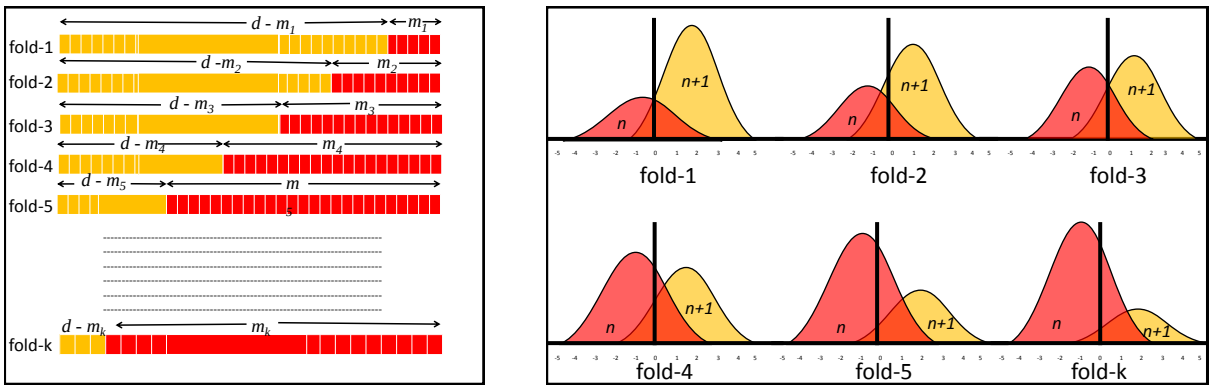
The sentiment analysis approaches based on semantic orientation use different semantic dictionaries (lists of sentiment words/lexical items with their semantic orientation or sentiment scores) to determine each individual word's semantic orientation. The range of the sentiment scores assigned to words in these dictionaries varies considerably. For instance, Taboada et al. (2011) used a dictionary with a sentiment score range between -5 and $+5$ whereas Esuli and Sebastiani (2007) has positive and sentiment words with scores between 0 and 1. Table 1 shows the different semantic scores for some common sentiment words in a number of recent semantic dictionaries[1]. In addition, the aggregation operations involved also vary, and do not always have straightforward semantic interpretations (for example, sentiment negation is achieved in some systems but inverting the score polarity, and in others by shifting the value towards zero). Comparing the outputs of such systems, or evaluating them against a gold standard, is therefore, very challenging.

2.2 The Precision vs. Recall Curve

The use of graphical representations to visualise classifer performance is well-established. The Receiver Operation Characteristic (ROC) curve, originally used in signal detection theory (Egan, 1975), has also been adopted to visualise classifier performances in text classification. The ROC is created by plotting true positive rates (TPR) against false positive rates (FPR) at various thresholds, and the area under the curve has been used as a measure of accuracy in evaluation methods. More recently, researchers have used the Precision-Recall (PR) curve, which plots precision against the true positive rate, and taken the area under this curve as a measure of performance (West et al., 2014; Manning and Schütze, 1999; Raghavan et al., 1989). Both curves can be used to visualise classifier performance; however, PR curves produce a more informative visualisation, particularly for highly imbalanced data sets (Davis and Goadrich, 2006). Moreover, a PR curve is more useful for problems where one class is considered to be more important than other classes. On the other hand, there are issues with PR curves too, for example unlike in ROC space it is complicated to interpolate two points in PR space. Furthermore, the area under a PR curve produces the arithmetic mean, whereas the also commonly used f-score is the harmonic mean of precision and recall[2]. However, these issues do not affect this work as in our calibration method we only use visualisation of the PR curve to set values for boundaries of sentiment classes.

[1]Bing Liu's opinion lexicon: www.cs.uic.edu/~liub/; Harvard General Inquirer: www.wjh.harvard.edu/~inquirer/; Vader Sentiment: github.com/cjhutto/vaderSentiment/tree/master/vaderSentiment; SentiWordNet: www.sentiwordnet.isti.cnr.it/; SenticNet: www.sentic.net/downloads/; Taboada et al. (2011)'s lexicon kindly made available by the authors for this research.

[2]Such issues can be mitigated by plotting a Precision-Recall-Gain curve (Flach and Kull, 2015) and considering its associated area. However this is beyond the scope of this paper.



(a) Folds containing mixtures of 2 document classes

(b) Example histograms and cut-off points for different balances between n^{th} and $n+1^{th}$ classes in different folds

Figure 1: k-fold class mixtures, to produce PR curves for each cut-off candidate

3 A Calibration Method for Cut-off Values of Sentiment Classes

In this section, we introduce a calibration method for setting sentiment class cut-off values from numerical sentiment scores using learning-based techniques. We use a training data set to assign boundaries of sentiment classes, where the classes have a natural 'sentiment order'. Our method is inspired by the cross-validation method. We calculate upper and lower boundary values of each sentiment class at a time in sentiment order. For instance, in a three-class classification, we first calculate boundary values for *negative* (1^{st} class), then *neutral* (2^{nd} class) and then *positive* (3^{rd} class). We then determine the optimal *cut-off* value between these two boundaries to delimit the classes.

To compute the cut-off value, first we reduce the problem of multi-classes and convert it into the standard binary class problem. That is, we consider the n^{th} order class and the $(n+1)^{th}$ order class to compute the cut-off values between those two classes. We select documents belonging to the n^{th} and $(n+1)^{th}$ classes from the training dataset and run our semantic classifier over these two sets. As a result, we get a set of numerical scores, one for each document in each class. We consider the maximum score for the n^{th} class, Max_n, and the minimum score for the $(n+1)^{th}$ class, Min_{n+1}. The cut-off value, $C_{n/n+1}$, for those two classes should lie between these two scores[3]. We plot different PR curves for candidate cut-off values between these scores to determine the cut-off value which gives optimum performance.

For a given candidate cut-off value, the PR curve plots the classifier system's ability to classify using that cut-off as the class boundary, for different mixtures of the two classes. The data set is divided into k subsets (folds) with an equal number (d) of documents. We assume the data set is normally distributed. Each subset contains n^{th} class documents and $(n+1)^{th}$ class documents in different proportions. For example, the 1^{st} subset contains m_1 number of n^{th} class documents and $(d-m_1)$ number of $(n+1)^{th}$ class documents, the 2^{nd} subset contains m_2 number of n^{th} class documents and $(d-m_2)$ number of $(n+1)^{th}$ class documents, and the k^{th} subset contains m_k number of n^{th} class documents and $(d-m_k)$ number of $(n+1)^{th}$ class documents (see figure 1a). Each fold represents a different distribution of sentiment scores for the two classes (see figure 1b) and hence a different precision and recall score for each class for the given cut-off. We then calculate the macro-average precison and recall across the two classes; the PR curve plots these different precision/recall values for a single cut-off value across all the folds.

The best cut-off value produces high and almost equal values of precision and recall. Therefore, the PR curve of the best cut-off value lies to the top right hand corner of the graph as well as close to the diagonal line ($p=r$). We originally hoped that we could choose the best PR curve by visual inspection, but in practice, while this is sufficient to rule out many candidates, the final choice was also supported by additionally plotting average recall and precision for each PR curve.

Once the best cut-off value, $C_{n/n+1}$, has been

[3]Note that the classes score ranges may overlap — Max_n may be greater than Min_{n+1}.

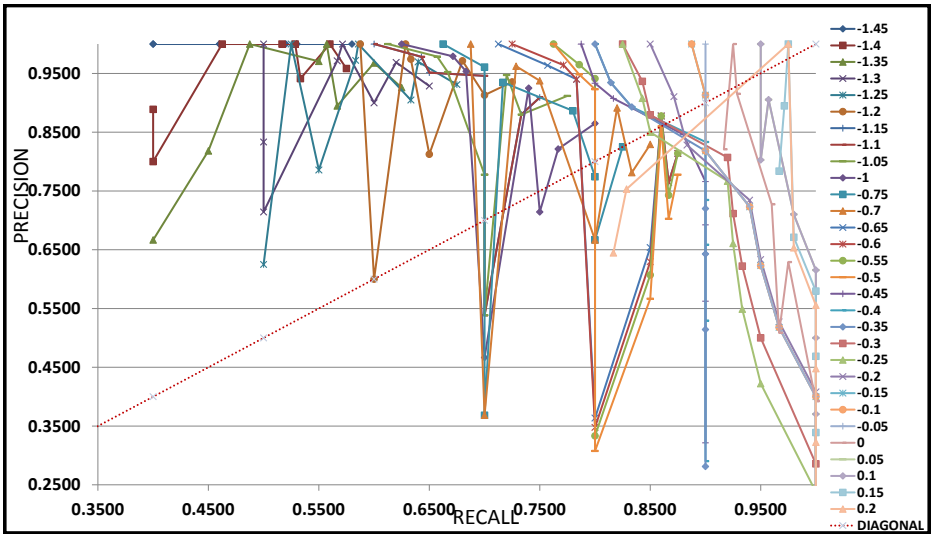


Figure 2: PR curves for all candidate cut-off values

established, we repeat the process for the other class boundaries ($C_{n+1/n+2}$ etc.). These cut-off values can then be used to map the numerical scores to classes in an optimal way. For example, in the three class *negative*, *neutral*, *positive* case, with classes 1, 2 and 3, we use $C_{1/2}$ as the boundary between *negative* and *neutral*, and $C_{2/3}$ as the boundary between *neutral* and *positive*, and classify as follows:

$$\mathrm{S}_i = \begin{cases} \textit{positive}, & \text{If } \textit{Tot}_i > C_{2/3} \\ \textit{neutral}, & \text{If } C_{1/2} < \textit{Tot}_i < C_{2/3} \\ \textit{negative}, & \text{If } \textit{Tot}_i < C_{1/2} \end{cases} \quad (1)$$

where S_i is the sentiment class of document i and $\textit{Tot}_i$ is the total sentiment score of the document i.

4 Experiments and Results

To test the above method, we performed an experiment with the *Galadriel* sentiment analysis system (Satthar, 2015) on a scaled dataset[4] used by Pang and Lee (2005). The dataset is a collection of movie reviews labelled with values of $0, 1, 2$. When analysed by the *Galadriel* system, the documents in this dataset return scores ranging between -10 and $+25$. The purpose of this experiment was to show that by assigning optimal cut-off values for *Galadriel* scores according to this scaled dataset, we can map the system's output into this three-class system in a way which maximises its performance as a sentiment classifier.

[4] www.cs.cornell.edu/people/pabo/movie-review-data/

We selected 300 documents of approximately equal length from the dataset (100 documents for each scale value in an approximately normal distribution). First we divided the dataset into two parts, one for training and other for testing. We used 240 documents (80 documents from each scale) as our training set. First, we computed boundaries for the *scale-0* class, then for the *scale-1* class and finally for the *scale-2* class. Since *scale-0* is the lowest class it is not necessary to compute the lower boundary for *scale-0*. To determine the upper boundary of the *Galadriel* score for *scale-0*, the cut-off value of the *Galadriel* score between *scale-0* and *scale-1* needed to be computed. For this, we used our *scale-0* and *scale-1* training documents (160 documents). We found that the maximum normalised *Galadriel* score for *scale-0* documents was $+0.17$ and minimum *Galadriel* score for *scale-1* documents was -1.41 (rounded up to two decimals). Therefore, we set up candidate cut-off values (C_i) between -1.45 and $+0.2$ in an equal interval of 0.05, i.e., -1.45, -1.40, -1.35, -1.30, -1.25, -1.20, -1.15, -1.10, -1.05, -1.00, -0.05, 0.00, $+0.5$, $+0.1$, $+0.15$, $+0.2$. Then, for each candidate cut-off value, we calculated precision and recall value for 5 sub training data sets, each subset containing a mixture of 35 *scale-0* and *scale-1* documents. For each cut-off values (C_i) precision and recall values were calculated for *scale-0* class and *scale-1* class. Then the precision and recall values were summarised by taking macro average of both classes' values. Finally, we had 5 pairs of precision and recall values for each of our 28 candidate

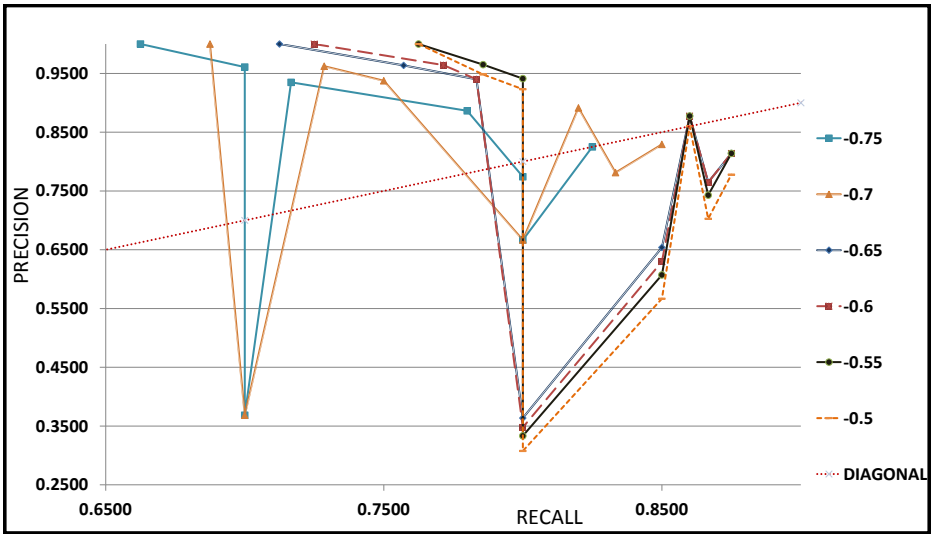


Figure 3: Most appropriate PR curves

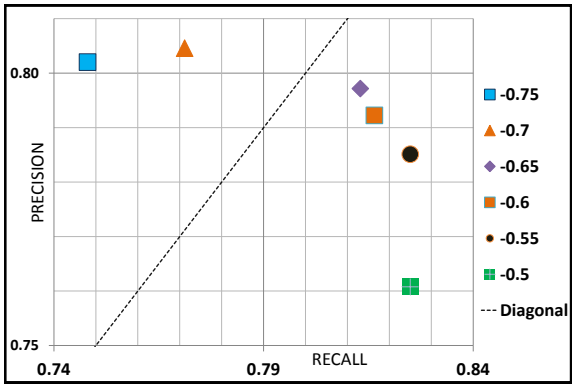


Figure 4: Average of Precision and Recall values

Cut-off values	Recall	Precision	F-score
-0.75	0.7480	0.8020	0.7741
-0.70	0.7712	0.8046	0.7875
-0.65	0.8131	0.7972	0.8050
-0.60	0.8164	0.7922	0.8042
-0.55	0.8250	0.7851	0.8046
-0.50	0.8250	0.7608	0.7916

Table 2: Average Precison, Recall and F-score measures for candidate cut-off values

cut-off values. Figure 2 shows the resulting 28 different PR curves.

The ideal cut-off value will have a PR curve as close to the diagonal, and as far towards the top right corner as possible. As can be seen in figure 2, although the general trend is for all the curves to be in the top right half of the graph, many of them deviate significantly from the diagonal line. We focused on the six curves closest to the diagonal (by visual inspection), shown in figure 3, for further analysis.

The 6 candidate cut-off values remaining after this step are -0.75, -0.70, -0.65, -0.60, -0.55 and -0.50. The PR curves of those values lie closest to the diagonal line, and largely in the upper right corner. Thus we concluded that one of those 6 test values is the optimal cut-off value $C_{0/1}$ for *scale-0* and *scale-1* classes. Looking more closely, we can see that the PR curves for -0.65,-0.60, -0.55 and -0.5 lie noticeably closer to the top right-hand corner compared to the PR curves for -0.75 and -0.70. We therefor discard these two, but the remaining curves track each other very closely — too closely for visual discrimination. We therefore calculated the (macro-)average precision and recall values of each cut-off value and plotted these in a scatter plot(figure 4). From this plot, we concluded that the best cut-off value for *scale-0* and *scale-1* classes is -0.65.

To validate this cut-off value, we also compared f-scores for the candidate cut-off values from these macro-averaged recall and precision values. We only considered the candidate values used in figure 3 as the remaining cut-off values had already been rejected. Table 2 also shows these numbers for the different candidate cut-off values. The f-score of the cut-off value -0.65 has the maximum value.

Similarly, the cut-off value $C_{1/2}$ for *scale-1* and *scale-2* classes were computed with an optimal value of $+1.05$.

Document Scales	Calibrated system			Uncalibrated system		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
scale-0	0.9375	0.7500	0.8333	0.6522	0.7500	0.6977
scale-1	0.7619	0.8000	0.7805	0.3333	0.0500	0.0870
scale-2	0.7826	0.9000	0.8372	0.5294	0.9000	0.6667
Macro-average	0.8273	0.8167	0.8220	0.5050	0.5667	0.4838

Table 4: Comparing performance measures calculated by the calibrated and uncalibrated versions of *Galadriel*.

Galadriel scores of documents	Scaled documents		
	0	**1**	**2**
$-0.65 > \text{Gal}_i$	15	1	0
$-0.65 < \text{Gal}_i < +1.05$	3	16	2
$+1.05 < \text{Gal}_i$	2	3	18

Table 3: Confusion Matrix for the classification

5 Evaluation of the Calibrated System

In order to demonstrate the effect of the calibration process, we evaluated the calibrated *Galadriel* system against Pang and Lee (2005)'s dataset and compared this with evaluation of the uncalibrated version. For this evaluation, we selected 50 random unseen test documents from the dataset and analysed them using *Galadriel*, giving numerical scores for each document as its output. The output scores were classified according to *Galadriel* cut-off values -0.65 ($C_{0/1}$) and $+1.05$ ($C_{1/2}$). Table 3 shows the resulting confusion matrix. It is interesting to note that this optimum score range for the *neutral* class is quite small in comparison to the total score range of the system (1.70 out of 30), and also not balanced around zero.

Table 4 shows precison, recall and f-score results for each class and overall macro-average results, for both the calibrated system and the uncalibrated system, which maps sentiment scores simply on the basis of their sign (negative, zero or positive). The effect of calibrating is to increase the macro-averaged f-score from 0.48 to 0.82. Moreover, the calibrated system gives overall macro-averaged mean absolute error (MAE) of 0.2167 whereas the uncalibrated system shows 0.5166.

6 Conclusion

This paper presented a novel calibration method to transform numerical sentiment scores into fixed ordered classes. This method uses corpus-based evaluation techniques, as widely used in supervised machine learning approaches, calibrating a system using gold standard labelled data. The effect is to optimise a continuous sentiment analysis system for the discrete classification model represented by the gold standard data. The calibrated system can then be evaluated and compared with other systems by using additional unseen gold standard data for the same model, or applied to new data assumed to follow the same model, with the confidence provided by the evaluation results. The availability of a general calibration method also means that the same system can be calibrated independently for different classification tasks as required.

We also presented a comparison between the performance of a calibrated system and the corresponding uncalibrated system, where sentiment scores are mapped into classes based solely on their sign, and showed that calibration can provide a substantial increase in performance. Although the uncalibrated system might be considered a poor baseline for comparison, it is worth bearing in mind that it is a simple model such as this which often guides the assignment of lexical semantic orientation scores such as those given in table 1. The effectiveness of calibration is a measure of the extent to which the document analysis process as a whole deviates from the simple lexical model, in a way that is difficult to capture by other means, and reveals interesting biases in the way the process maps sentiment onto scores.

In future work, we hope to look at automating the process of selecting the best PR curve, so that the entire calibration process is essentially automatic.

References

Mohamed Aly. 2005. Survey on multiclass classification methods. *Neural Networks* 19:1–9.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Evaluation measures for ordinal regression. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*. IEEE, pages 283–287.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '06, pages 233–240. https://doi.org/10.1145/1143844.1143874.

James P. Egan. 1975. *Signal detection theory and ROC analysis*. Academic Press, New York.

Andrea Esuli and Fabrizio Sebastiani. 2007. Sentiwordnet: A high-coverage lexical resource for opinion mining. *Evaluation* pages 1–26.

Peter Flach and Meelis Kull. 2015. Precision-recall-gain curves: Pr analysis done right. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 838–846. http://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right.pdf.

Lisa Gaudette and Nathalie Japkowicz. 2009. Evaluation methods for ordinal classification. In *Canadian Conference on Artificial Intelligence*. Springer, pages 207–210.

Raymond Hsu, Bozhi See, and Alan Wu. 2010. Machine learning for sentiment analysis on the Experience project. Accessed on July 31, 2017. http://cs229.stanford.edu/proj2010/HsuSeeWu-MachineLearningForSentimentAnalysis.pdf.

Moontae Lee and Patrick Grafe. 2010. Multiclass sentiment analysis with restaurant reviews. Accessed on July 31, 2017. https://nlp.stanford.edu/courses/cs224n/2010/reports/pgrafe-moontae.pdf.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1–18. http://www.aclweb.org/anthology/S16-1001.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2Nd International Conference on Knowledge Capture*. ACM, New York, NY, USA, K-CAP '03, pages 70–77. https://doi.org/10.1145/945645.945658.

Prabu Palanisamy, Vineet Yadav, and Harsha Elchuri. 2013. Serendio: Simple and practical lexicon based approach to sentiment analysis. In *proceedings of Second Joint Conference on Lexical and Computational Semantics*. Citeseer, pages 543–548.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '05, pages 115–124. https://doi.org/10.3115/1219840.1219855.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pages 79–86.

Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics* 3(2):143–157.

Vijay Raghavan, Peter Bollmann, and Gwang S. Jung. 1989. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.* 7(3):205–229. https://doi.org/10.1145/65943.65945.

Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. 2016. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management* 52(1):5–19.

F Sharmila Satthar. 2015. Modelling so-cal in an inheritance-based sentiment analysis framework. In *OASIcs-OpenAccess Series in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, volume 49.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2):267–307.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*. ACL, pages 417–424.

Muqtar Unnisa, Ayesha Ameen, and Syed Raziuddin. 2016. Opinion mining on twitter data using unsupervised learning technique. *International Journal of Computer Applications* 148(12).

Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *arXiv preprint arXiv:1409.2450* .

Taras Zagibalov and John Carroll. 2008. Automatic seed word selection for unsupervised sentiment classification of chinese text. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 1073–1080.