Optimising Agile Social Media Analysis

Thomas Kober
Department of Informatics
University of Sussex
Brighton, UK
t.kober@sussex.ac.uk

David Weir
Department of Informatics
University of Sussex
Brighton, UK
d.j.weir@sussex.ac.uk

Abstract

Agile social media analysis involves building bespoke, one-off classification pipelines tailored to the analysis of specific datasets. In this study we investigate how the DUALIST architecture can be optimised for agile social media analysis. We evaluate several semi-supervised learning algorithms in conjunction with a Naïve Bayes model, and show how these modifications can improve the performance of bespoke classifiers for a variety of tasks on a large range of datasets.

1 Introduction

Natural Language Processing (NLP) on large social media datasets has emerged as a popular theme in the academic NLP community with publications ranging from predicting elections, e.g. (Tumasjan et al., 2010; Marchetti-Bowick and Chambers, 2012), to forecasting box-office revenues for movies, e.g. (Asur and Huberman, 2010) and anticipating the stock market, e.g. (Bollen et al., 2011; Si et al., 2013). More recently, Opinion Mining and Sentiment Analysis on large social media datasets have received an increasing amount of attention outside academia, where a growing number of businesses and public institutions seek to gain insight into public opinion. For example, companies are primarily interested in what is being said about their brand and products, while public organisations are more concerned with analysing reactions to recent events, or with capturing the general political and societal Zeitgeist. The social network Twitter has been a popular target for such analyses as the vast majority of tweets are publicly available, and easily obtainable via the Twitter API[1], which conveniently enables the harnessing of a large number of real-time responses to any user-defined keyword query.

In this paper we are concerned with what we call *agile social media analysis*, which is best illustrated with an example. Imagine that a political scientist wants to investigate reactions on Twitter to a speech given by British Prime Minister David Cameron the previous night. She uses an application which allows her to query the Twitter API in order to gather a dataset, and to interactively design classifiers, tailored to specific tasks. For her analysis, she starts searching for "Cameron", which inevitably will retrieve a large number of irrelevant tweets, e.g. those referring to Cameron Diaz. Her first goal therefore is to filter out all of those unrelated tweets, for which she requires a bespoke classifier that will only be used for *this single task*. In order to create such a classifier, she first needs to annotate a gold standard evaluation set which is randomly sampled from the initially retrieved tweets. While labelling the first few tweets for the evaluation set, she starts to build a picture of the range of topics being discussed on Twitter that night. She notices that a considerable proportion of tweets appears to be talking about David Cameron's personality. Many of the others appear to be about two specific topics mentioned in the speech: tax policy and the EU referendum. After training a classifier to perform relevancy classification, she therefore decides to create another one-off classifier to divide the relevant tweets into the three categories, "personality", "tax policy" and "EU referendum". To conclude her analysis, she creates three more bespoke classifiers to perform Sentiment Analysis on each of the three subsets *separately*.

A crucial aspect of performing agile social media analysis is the direct interaction with the data, through which the analyst gains a sense of what the discourse is about. It furthermore enables her to better tailor her analysis to the collected

[1] http://dev.twitter.com/

Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2015), pages 31–40,
Lisboa, Portugal, 17 September, 2015. ©2015 Association for Computational Linguistics.

data. DUALIST introduced the framework which enables non-technical analysts to design bespoke classifiers by labelling documents and features through active learning, with only a few minutes of annotation effort (Settles, 2011; Settles and Zhu, 2012). Wibberley et al. (2013) and Wibberley et al. (2014) showed that the DUALIST architecture can successfully be used for performing ad-hoc analyses in an agile manner.

The remainder of this paper is organised as follows: in Section 2 we more generally introduce agile social media analysis, followed by the description of the datasets we use in our empirical evaluation in Section 3. Section 4 describes our approach alongside related work and Section 5 presents our experiments and discusses our findings. In Section 6 we give an overview of future work and we conclude this paper in Section 7.

2 Agile Social Media Analysis

When beginning an analysis the social scientist has no predetermined plan of the specific content of her investigation. The reason is that there is limited appreciation for what is being discussed in advance of engaging with the data. Therefore, the process of annotating a gold standard evaluation set and a training set to create bespoke classifiers, also serves the purpose of exploring the data space.

After collecting a text corpus from Twitter, the analyst typically creates a tailored multi-stage classification pipeline to organise the heterogenous mass of data. As explained in the introductory scenario, the first stage often involves filtering irrelevant tweets, since keyword queries are purposefully kept broad to minimise the risk of missing relevant aspects of a discussion. The following stages are completely dependent on the extracted content — target categories are not known upfront, but are determined while interacting with the data. Each stage in this pipeline requires the annotation of a gold standard evaluation set and the training of a bespoke classifier to perform the categorisation. The tweets for the gold standard set are randomly sampled from the available data, whereas the creation of the classifier is guided by active learning to accelerate the training process (Settles, 2009). The two kinds of labelling tasks have intrinsic beneficial side-effects that support the analyst's investigation. When annotating a gold standard set, the social scientist is able to explore the data and gather ideas for further analyses. The training of a bespoke classifier enables the analyst to quickly test whether the algorithm has the capability to divide the data into the target categories. This is possible because the system is able to provide instant feedback on how well the current classifier is performing on the evaluation set, and allows the social scientist to "fail fast". This has the benefit of being able to quickly define new target categories which better match the data.

From a Machine Learning perspective, agile social media analysis poses a number of distinct challenges. The labelled data for any classification task can contain a considerable amount of noise as the dataset is not labelled and validated by a team of experienced annotators in month-long efforts, but in short sessions by a single analyst. Furthermore, for most downstream classification tasks, the input dataset often is the product of one or more preceding classifiers. Therefore, there is no guarantee that a given tweet is actually relevant to the current analysis.

The small amount of labelled data together with the large amount of unlabelled data raise the issue of how to best make effective use of the vast number of unlabelled tweets. We investigate this problem from two complementing angles. On the one side we enhance our current semi-supervised learning algorithm with several simple modifications. On the other side, we compare the adjusted algorithms with various other semi-supervised learning algorithms that aim to leverage the information in the unlabelled data in a different way. We furthermore examine whether we can improve the classifier by extending its language model to include bigrams and trigrams.

3 Datasets

We evaluate our experiments on 24 Twitter datasets that have been collected by social scientists for a number of real-world analyses (Bartlett and Norrie, 2015; Bartlett et al., 2014b; Bartlett et al., 2014a). The Twitter datasets represent a diverse range of possible applications of agile social media analysis. Some are focused on "Twitcidents"[2] during political debates or speeches (*boocheer, cameron 1-3, clacton, clegg, debate 1-2, farage, immigr, miliband 1-2, salmond*). Three

[2] "A major incident provoking a firestorm of reactions on Twitter", see `http://www.urbandictionary.com/define.php?term=Twitcident`

datasets are concerned with reactions to the inquest following the death of Mark Duggan in London 2013 (*duggan 1-3*), and the remaining ones investigate topics such as the winter floods in many coastal regions in the South of England, throughout late 2013 and early 2014 (*flood 1-2*), misogyny (*misogyny, rape*), extremism (*isis 1-3*) and oil drillings in the arctic (*shell*). The Twitter datasets are drawn from different stages of the processing pipeline, which means that some datasets consist of the unprocessed tweets matching an initial keyword query while others have already been processed by one or more preceding steps in the pipeline. For example, the *shell* and *flood-1* datasets are the result of querying the Twitter API, whereas the *duggan-1* dataset has already been cleared of irrelevant tweets, and tweets only containing news links, in two separate preceeding stages of the processing pipeline. We furthermore evaluate our implementations on 2 commonly used NLP benchmark datasets, 20 Newsgroups (Lang, 1995), henceforth "*20news*", as an example Topic Classification dataset, and Movie Reviews (Maas et al., 2011), henceforth "*reviews*", as an example Sentiment Analysis dataset.

Table 1 highlights the extreme imbalance between labelled and unlabelled data and the corresponding differences in vocabulary size. In the Twitter datasets, $|V_{\mathcal{L}}|$ is usually one order of magnitude smaller than $|V_{\mathcal{L}\cup\mathcal{U}}|$. In comparison, the disparity in vocabulary size between labelled and unlabelled data in the *reviews* corpus is less than a factor of two. The difference is more extreme when looking at the actual amounts of labelled and unlabelled data, where the Twitter datasets often contain two orders of magnitude more unlabelled data than labelled data. Furthermore, the disparity in number of labelled documents between the Twitter datasets and the NLP benchmark corpora usually is one to two orders of magnitude. Where the *20news* dataset contains more than 10k labelled documents and the *reviews* dataset even 25k labelled instances, the Twitter datasets rarely contain more than a few hundred labelled tweets.

4 Approach & Related Work

The DUALIST architecture represents the general framework for performing agile social media analysis by combining active learning, semi-supervised learning, a Naïve Bayes text classifier and a graphical user interface into an application. A human annotator iteratively labels new tweets and terms in tweets which the active learning algorithm identifies as being most beneficial for annotation. The flexibility to label instances and individual words perhaps is the most important reason why effective classifiers can be created with only a few minutes of labelling effort. To leverage the collective information of the labelled and unlabelled data, DUALIST executes a single iteration of the Expectation-Maximization algorithm (Settles, 2011). In this paper we focus on the Naïve Bayes classifier and the semi-supervised learning algorithm and leave an investigation of the active learning component — and especially the feature labelling — for future work.

4.1 Naïve Bayes

Naïve Bayes fulfills the most important requirements for agile social media analysis: it is fast to train, proven to work well in the text domain despite its often violated independence assumptions, and is easily extensible with semi-supervised learning algorithms such as Expectation-Maximization due to its generative nature (Domingos and Pazzani, 1997; McCallum and Nigam, 1998; Nigam et al., 2000). The goal of classification is to find the class $c \in C$ that is most likely to have generated document d, which Naïve Bayes estimates as:

$$c = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i=1}^{N_D} P(w_i \mid c) \qquad (1)$$

where $P(w_i \mid c)$ is the conditional probability of word w_i occurring in a document of class c, containing N_D words in a given labelled dataset $\mathcal{L}$.

4.2 Which Naïve Bayes?

There are several distinct flavours of the Naïve Bayes model, with different model types being better suited for some tasks and data characteristics than others. One major distinction is whether a Multinomial or Bernoulli event model is used. The former incorporates term frequency information into the model whereas the latter only uses term occurrence information. It has been shown that the Multinomial model usually performs better in the Topic Classification domain (McCallum and Nigam, 1998). However, Manning et al. (2008) highlight that the Bernoulli model tends to work better for short texts. Interestingly, a variant of the Multinomial event model that only uses binary

Name	$\mathcal{T}$	$\lvert C\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert V_{\mathcal{L}}\rvert$	$\lvert V_{\mathcal{L}\cup\mathcal{U}}\rvert$	Name	$\mathcal{T}$	$\lvert C\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert V_{\mathcal{L}}\rvert$	$\lvert V_{\mathcal{L}\cup\mathcal{U}}\rvert$
20news	TC	20	11314	-	130107	130107	*flood-1*	TC	2	530	116123	2176	72004
boo-cheer	SA	3	1665	436305	7092	104477	*flood-2*	TC	4	1615	39327	5043	25326
cameron-1	TC	2	205	33561	1491	33234	*immigr*	TC	2	210	425425	1098	171195
cameron-2	TC	4	320	867637	1317	294169	*isis-1*	SA	3	322	19378	2123	32242
cameron-3	SA	3	502	303868	1858	122372	*isis-2*	TC	2	827	107310	2549	51444
clacton	SA	3	930	147493	2785	59990	*isis-3*	TC	2	602	56928	1859	29287
clegg	SA	3	500	9597	3280	8349	*miliband-1*	SA	3	927	36335	3378	19728
debate-1	SA	3	306	31993	917	10987	*miliband-2*	SA	3	449	35786	2092	19785
debate-2	TC	5	123	31993	482	10984	*misogyny*	TC	2	215	119078	1131	89474
duggan-1	TC	3	475	86749	1376	26382	*rape*	TC	3	746	108044	3908	78757
duggan-2	TC	4	1086	53440	2609	15760	*reviews*	SA	2	25000	50000	74849	124255
duggan-3	TC	3	401	86749	1283	26385	*salmond*	SA	3	228	55899	1171	14464
farage	SA	3	2614	9794	5305	8349	*shell*	TC	2	221	50065	1196	60815

Table 1: Datasets: $\mathcal{T}$=Task, where TC=Topic Classification; SA=Sentiment Analysis; $|C|$ = number of labels; $\mathcal{L}$=Labelled data, $|\mathcal{L}|$=amount of Labelled data; $\mathcal{U}$=Unlabelled data, $|\mathcal{U}|$=amount of Unlabelled data; $|V_{\mathcal{L}}|$=Vocabulary size of the labelled data; $|V_{\mathcal{L}\cup\mathcal{U}}|$=Vocabulary size of the labelled and unlabelled data

counts instead of the full frequency information has been shown to outperform the standard Multinomial model, and the Bernoulli model, for a variety of tasks (Metsis et al., 2006; Wang and Manning, 2012).

Instead of the commonly used Laplacian (add-1) smoothing, we use a simple heuristic that adjusts the additive smoothing term, depending on the number of observed tokens and the overall vocabulary size, for every dataset individually. Instead of adding 1, we add $\frac{1}{10^k}$, and normalise appropriately afterwards. We defined $k = \left\lfloor \frac{\log |V_{\mathcal{L}\cup\mathcal{U}}|}{\log |T_{\mathcal{L}}|} \right\rfloor$, where $|V_{\mathcal{L}\cup\mathcal{U}}|$ is the total size of the vocabulary and $|T_{\mathcal{L}}|$ is the number of tokens in the labelled data. This approach re-distributes probability mass from observed words to unknown ones less aggressively than add-1 smoothing. We refer to this heuristic as Lidstone-Tokens (LT) smoothing and compare it to add-1 smoothing in a supervised learning scenario.

4.3 Semi-supervised Learning

In these experiments we examine the performance of three semi-supervised learning algorithms — Expectation-Maximization and two more recently proposed algorithms, Semi-supervised Frequency Estimate (Su et al., 2011), and Feature Marginals (Lucas and Downey, 2013).

4.4 Expectation-Maximization

The starting point for the Expectation-Maximization (EM) algorithm is an initial model instance from the labelled data $\mathcal{L}$, which can be obtained in a number of ways. A common approach is to train a Naïve Bayes classifier on the available labelled documents. DUALIST introduced an alternative using the labelled *features*, whose term frequencies are incremented by a pseudo-count, which was found to be more effective for an active learning scenario (Settles, 2011). In order to factor out the effect of active learning and to better study our modifications on datasets without any labelled features, we are using the labelled instances to initialise EM.

The EM algorithm first produces probabilistic class predictions for the unlabelled data $\mathcal{U}$, representing the "E-Step" and subsequently re-estimates the model parameters on all available data $\mathcal{L} \cup \mathcal{U}$ in the "M-Step". These two steps can be repeated until convergence, although for efficiency reasons, DUALIST only performs a single iteration. Furthermore, given the enormous difference in amounts of labelled and unlabelled data, documents in $\mathcal{U}$ are assigned a smaller weight than data in $\mathcal{L}$ in order to not drown out the information learnt from the labelled data. A common approach is to assign every instance in $\mathcal{U}$ a weight of $\alpha = 0.1$, henceforth "EM-CWF"[3] (Nigam et al., 2000; Settles, 2011). In a typical practical application, from which most of our datasets are drawn, we observe only a few hundred labelled documents but several tens or hundreds of thousands of unlabelled instances. In these circumstances, it can be hypothesised that a weight of $\alpha = 0.1$ would be too high, and the unlabelled data would outweigh the labelled data by one to two orders of magnitude. We therefore assign tweets in $\mathcal{U}$ a weight of $\alpha = \frac{|\mathcal{L}|}{|\mathcal{U}|}$, where $|\mathcal{L}|$ represents the number of labelled documents and $|\mathcal{U}|$ represents the number of unlabelled documents. We refer to this weighting scheme as "Proportional Weight Factor" (PWF).

4.5 Semi-supervised Frequency Estimate

The Semi-supervised Frequency Estimate (SFE) algorithm leverages the information $P(w)$ over the combined amount of labelled and unlabelled

[3] CWF means "Constant Weight Factor". For all of our experiments EM-CWF refers to the specific case with $\alpha = 0.1$.

data, to scale the class-conditional probabilities learnt from $\mathcal{L}$. Hence, the probability mass is redistributed according to a word's overall prevalence in the corpus. Unlike the EM algorithm, SFE only requires a single pass over the data to adjust the model parameters and is thus better able to scale to large amounts of unlabelled data. SFE does not need the adjustment of additional hyperparameters such as the weighting of probabilistically labelled documents in $\mathcal{U}$.

4.6 Feature Marginals

The Feature Marginals (FM) algorithm also uses the information of $P(w)$ over the labelled and unlabelled data to scale the class-conditional probabilities estimated from the training set. In addition, FM re-distributes the probability mass of $P(w)$ according to the probability of a token in $\mathcal{L}$ occurring in either class. Lucas and Downey (2013) found that their model is especially effective in estimating probabilities for words that have not been seen in the labelled data. In its current form, FM does not generalise to multi-class problems, we therefore perform one-vs-rest classification for datasets with more than two classes.

4.7 Usefulness of Unlabelled Data

Previous work has shown that unlabelled data can be leveraged to create superior models (Chawla and Karakoulas, 2005). The DUALIST framework adopts the assumption that by exploiting semi-supervised learning techniques, a more effective model can be built than by supervised learning alone. We examine whether the benefits of semi-supervised learning hold for the distinctive characteristics in our Twitter datasets.

4.8 Feature Extraction — Unigrams, Bigrams or Trigrams?

We investigate whether classifier performance can be improved by including bigram and trigram features. Wang and Manning (2012) showed that bigram features are especially beneficial for a more complex task such as Sentiment Analysis, but also consistently improve performance on Topic Classification problems for supervised learning settings.

5 Experiments & Discussion

All datasets we use have pre-defined training/testing splits. We tokenise the documents, but do not perform any other pre-processing such as stemming, URL normalisation or stopword removal. All documents are represented as simple bag-of-words vectors. We report micro-averaged F1-Scores for all experiments. When investigating the effect of unlabelled data, we randomly sample 1k, 5k, 10k, 25k, 50k, 100k unlabelled tweets, or use all available unlabelled data. As baseline we use EM-CWF — MNB add-1, which reflects the text classifier and semi-supervised learning algorithm used in DUALIST, with the difference that we use the labelled documents instead of the labelled features for initialising EM. This is to isolate the effects of Naïve Bayes and EM, and to factor out the contributions of active learning. We compare our results in terms of absolute F1-Score gain/loss in comparison to our baseline, or present F1-Score performance trajectories.

5.1 Parameterisation and Selection of the Naïve Bayes Event Model

As Figure 1 shows, Lidstone-Tokens smoothing performs better than add-1 smoothing on 18 out of 26 datasets, and improves F1-Score by 2.5% on average across all datasets, in a supervised learning scenario. We therefore adopt it for all further experiments. We furthermore drop the standard Multinomial Naïve Bayes model and only adopt the binary MNB and the Bernoulli Naïve Bayes (BNB) models for future comparisons, as we found them to be superior to the standard Multinomial model. Our findings are consistent with previously published results of Wang and Manning (2012), and Metsis et al. (2006), who report that binary MNB works better than the standard Multinomial model for a variety of Topic Classification and Sentiment Analysis tasks. Our results also agree with Manning et al. (2008) who found the Bernoulli event model to be a competitive choice for short text classification. For all experiments we use all combinations of binary MNB and BNB together with the three semi-supervised learning algorithms introduced in the previous section — except for BNB + FM, which we found to significantly underperfom the other combinations.

5.2 Semi-supervised Learning Algorithms Comparison

As Figure 2 shows, there are only two datasets (*clacton* and *isis-2*), where the EM-CWF — MNB add-1 baseline outperforms the other semi-supervised learning algorithms. On the other

Name	EM-*M*	EM-*B*	SFE-*M*	SFE-*B*	FM	MNB	EM-C	Name	EM-*M*	EM-*B*	SFE-*M*	SFE-*B*	FM	MNB	EM-C
20news	0.761	0.676	**0.779**	**0.779**	0.743	0.759	0.729	*flood-1*	**0.468**	0.368	0.451	0.456	0.381	0.381	0.382
boo-cheer	0.492	0.516	0.538	**0.539**	0.498	0.496	0.487	*flood-2*	0.669	**0.704**	0.669	0.669	0.683	0.683	0.62
cameron-1	0.781	**0.832**	0.815	0.827	0.814	0.808	0.712	*immigr*	0.956	0.91	0.951	0.948	0.953	**0.962**	0.91
cameron-2	**0.589**	0.529	**0.589**	0.58	**0.589**	0.585	0.451	*isis-1*	0.567	**0.573**	0.533	0.557	0.55	0.563	0.503
cameron-3	0.67	0.683	0.647	0.667	**0.7**	**0.7**	0.62	*isis-2*	0.751	0.755	0.734	0.722	0.758	0.753	**0.808**
clacton	0.52	0.483	0.513	0.513	0.517	0.513	**0.56**	*isis-3*	0.648	**0.667**	0.658	0.648	0.654	0.654	0.533
clegg	0.696	0.724	0.692	0.7	**0.736**	0.724	0.676	*miliband-1*	0.556	0.563	0.544	0.552	0.57	**0.574**	0.533
debate-1	0.627	**0.637**	0.597	0.61	0.63	0.626	0.5	*miliband-2*	0.69	0.69	**0.71**	0.7	0.69	0.7	0.6
debate-2	0.667	0.661	0.567	0.581	0.644	**0.684**	0.396	*misogyny*	**0.953**	**0.953**	0.938	0.949	**0.953**	**0.953**	0.888
duggan-1	0.639	**0.649**	0.634	0.634	0.637	0.634	0.526	*rape*	**0.895**	0.87	0.87	0.88	**0.895**	0.885	0.785
duggan-2	0.585	**0.603**	0.537	0.545	0.575	0.6	0.378	*reviews*	0.826	0.825	**0.831**	**0.831**	0.83	0.83	0.821
duggan-3	0.767	0.75	**0.773**	0.767	0.76	0.75	0.603	*salmond*	**0.69**	**0.69**	0.65	0.68	**0.69**	**0.69**	0.45
farage	0.718	**0.72**	0.698	0.712	0.67	0.716	0.688	*shell*	0.756	**0.798**	0.738	0.755	0.758	0.775	0.715

Table 2: Micro averaged F1-Score for all methods across all datasets. EM-*M*=EM-PWF — binary MNB LT; EM-*B*=EM-PWF — BNB LT; SFE-*M*=SFE — binary MNB LT; SFE-*B*=SFE — BNB LT; FM=FM — binary MNB LT; MNB=supervised binary MNB LT; EM-C=EM-CWF — MNB add-1; Boldfaced numbers mean top performance on the dataset.

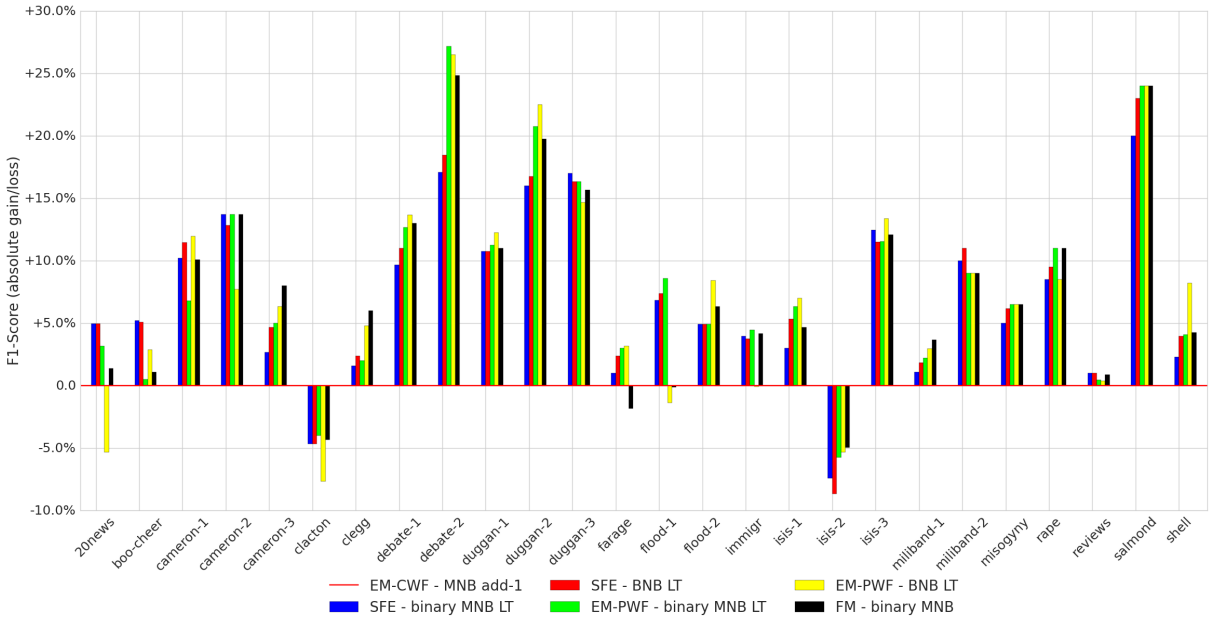


Figure 2: Semi-Supervised Learning algorithm comparison. The baseline is EM-CWF — MNB add-1. PWF refers to our EM weighting scheme. The new algorithms only failed to improve performance on 2 datasets. Our simple enhancements to NB smoothing and EM weighting (see Sections 4.1 and 4.3) improve an NB-EM combination considerably and make it competitive with SFE and FM.

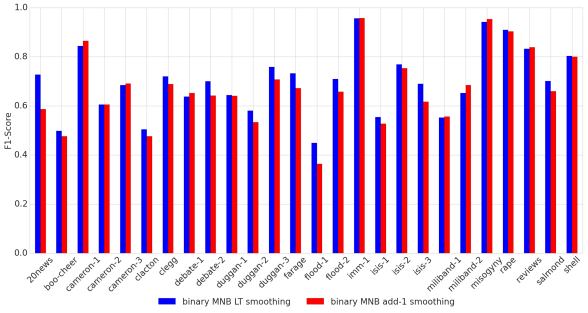


Figure 1: Overall Lidstone-Tokens smoothing achieves an average improvement of 2.5% across all datasets and improves performance on 18 out of 26 datasets. Performance gains are as large as 14% in absolute terms on the *20News* dataset, 8.5% on the *flood-1* dataset and 7.3% on the *isis-2* dataset. Both MNB models use binary counts.

hand, there is no single dominant algorithm that consistently outperforms the others (also see Table 2). Our results confirm that SFE and FM are superior to EM-CWF — MNB add-1 as was shown in the respective publications, and that their improvements can be leveraged for agile social media analysis. Interestingly, our simple modifications to EM improve its performance substantially, making it competitive with SFE and FM on our datasets. Our results furthermore highlight that considerable performance improvements can be gained for the commonly used combination of Naïve Bayes and Expectation-Maximization when their respective hyperparameters are optimised for the given dataset characteristics.

5.3 The Effect of Unlabelled Data

Table 2 shows that adding unlabelled data does not always improve performance. The supervised binary MNB classifier with Lidstone-Tokens smoothing is the top performing method on 6 out of 26 datasets. Only the EM-PWF — BNB LT combination is the top performing method more frequently. Figures 3a and 3b show that EM-CWF — MNB add-1 appears to be very sensitive to the amount of unlabelled data, whereas the other semi-supervised learning algorithms remain relatively stable under a growing amount of unlabelled data. Figure 3a highlights a prototyp-

ical case where adding unlabelled data up to a certain threshold improves performance, but *degrades* it when more is added. We observed this behaviour of EM-CWF — MNB add-1 on a number of datasets. Figure 3b shows that EM-PWF — binary MNB LT, FM — binary MNB LT and SFE — binary MNB LT do not make the most effective use of the unlabelled data, hence there is still potential for further improvement in these algorithms. Especially EM-PWF — binary MNB LT is perhaps scaling down the contributions of the unlabelled data too aggressively. This comes at the expense of not leveraging the full potential of the unlabelled documents, but has the advantage of improved stability across varying amounts of unlabelled data as our experiments show.

5.4 The Effect of Adding Bigrams and Trigrams

Contrary to our expectations, adding bigrams or trigrams produced mixed results and did not consistently improve performance on our datasets. An interesting observation is the different behaviour of the various semi-supervised algorithms. For example, adding trigrams improves EM-PWF — binary MNB LT by almost 10% on the *flood-1* dataset, whereas performance goes down by nearly 10% for SFE — binary MNB LT. The reverse effect can be observed on the *shell* dataset. Our findings are in contrast to published results by Wang and Manning (2012) who report that adding bigrams never degraded performance in their experiments. Figures 4a-4c highlight the inconsistent behaviour of adding bigrams or trigrams for three semi-supervised learning algorithms across all datasets[4]. We also ran our experiments with a purely supervised MNB classifier to factor out the effect of semi-supervised learning, which however, resulted in the same inconsistent behaviour (see Figure 4d). A closer investigation of the datasets suggests that the difference might be due to the idiosyncrasy of Twitter where opinions are commonly packaged into multi-word hashtag expressions, which frequently capture the sentiment of a tweet, but are treated as unigrams. For example, expressions such as "#CameronMustGo" and "#CareNotCuts" in the *boo-cheer* dataset, or "#NoSympathy" and "#PoliceMurder" in the *duggan-1* dataset, convey crucial sentiment information. The phenomenon is not exclusive to Sentiment Analysis, hashtag expressions frequently categorise a tweet, e.g. "#ArcticOil" in the *shell* dataset. Such topical information has already been leveraged in a number of previous works, e.g. Weston et al. (2014); Dela Rosa et al. (2011). Therefore, we hypothesise that the potential benefits of bigrams or trigrams cannot be leveraged as effectively for Twitter Sentiment Analysis datasets than for other datasets.

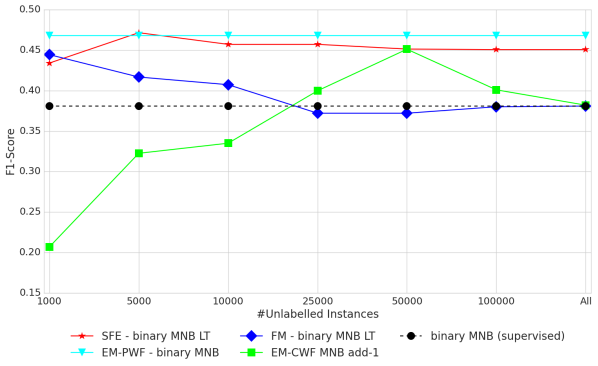
[4]Due to space reasons, we only show figures for the binary MNB variants — the results for the BNB variants are almost identical.

6 Future Work

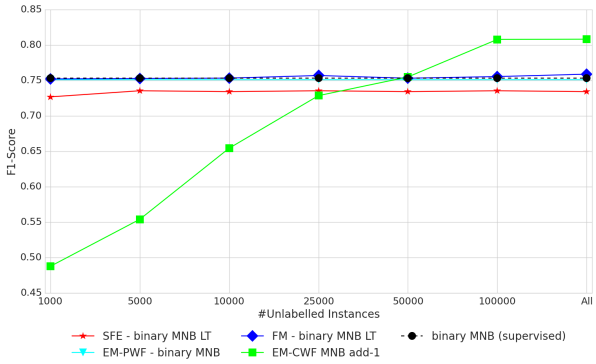
Our results created a multitude of directions for future research. We plan to investigate the reason behind the inconsistent performance of the semi-supervised learning algorithms across our datasets. We are interested whether it is specific dataset characteristics or particular hyperparameter configurations that cause e.g. EM-PWF — BNB LT to be the top performing algorithm on the *shell* and *duggan-2* datasets, but the worst performer on the *clacton* and *flood-1* datasets. Moreover, we seek to gain insight why adding bigrams or trigrams *improves* performance on a given dataset for one method, but degrades it for another. We also plan to study whether we can use the unlabelled data more effectively, e.g. by subsampling the unlabelled tweets by some criterion. The hypothesis is that there might be a subset of tweets in the unlabelled data which better aligns with the current analysis. We will furthermore examine whether the active learning process, and especially the feature labelling, can be improved in order to create more effective bespoke classifiers with less manual labelling effort. Lastly, we intend to investigate the role of opinionated multi-word hashtag expressions which not only convey topical information, but also express sentiment as we highlighted in the previous section. We therefore intend to assess whether we can leverage the sentiment information of hashtag expressions to improve Sentiment Analysis on our Twitter datasets.

7 Conclusion

In this paper we highlighted the demand for being able to quickly build bespoke classifier pipelines when performing agile social media analysis in practice. We considered different Naïve Bayes event models in conjunction with various semi-supervised learning algorithms on a large range of datasets. We showed that SFE and FM outperform

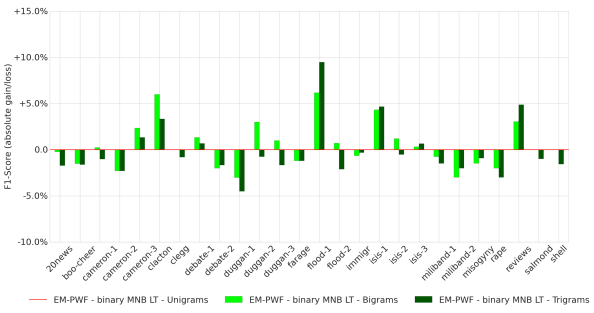


(a) The effect of unlabelled data on the *flood-1* dataset. While EM-PWF — binary MNB LT and SFE — binary MNB LT are relatively stable with increasing amounts of unlabelled data, EM-CWF — MNB add-1 displays its frequently observed "peak-behaviour", where adding unlabelled data would improve performance until a threshold is reached, after which performance degrades again. FM — binary MNB LT shows the opposite effect, where performance decreases in the beginning and then slightly recovers with more unlabelled data.

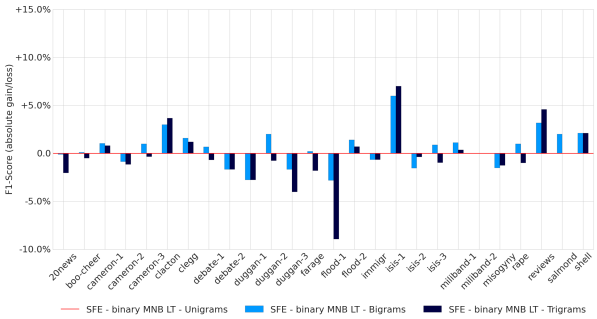


(b) The effect of unlabelled data on the *isis-2* dataset. While the performance of EM-CWF — MNB add-1 increases steadily with more unlabelled data, EM-PWF — binary MNB LT, FM — binary MNB LT and SFE — binary MNB LT remain very stable with increasing amounts of unlabelled data. These results suggest that there is further room for improvement in the latter algorithms to make more effective use of the unlabelled data.

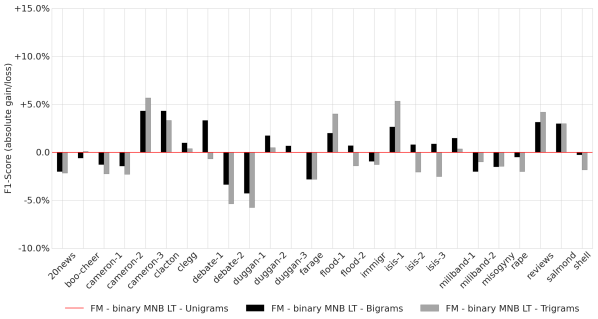
Figure 3: Micro averaged F1-Score over the number of unlabelled instances. The baseline is a supervised binary MNB LT classifier. To reduce clutter, we only present the binary MNB variants of EM-PWF, FM and SFE.



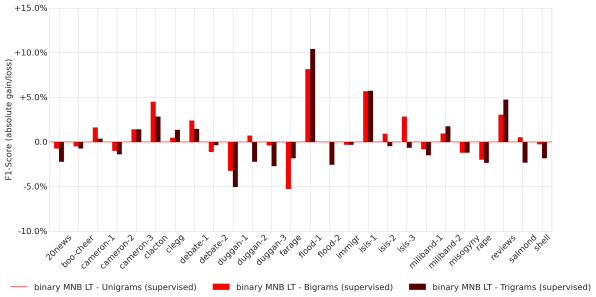
(a) The effect of adding bigrams and trigrams in comparison to a unigram baseline for the EM-PWF — binary MNB LT algorithm.



(b) The effect of adding bigrams and trigrams in comparison to a unigram baseline for the SFE — binary MNB LT algorithm.



(c) The effect of adding bigrams and trigrams in comparison to a unigram baseline for the FM — binary MNB LT algorithm.



(d) The effect of adding bigrams and trigrams in comparison to a unigram baseline for the supervised binary MNB algorithm.

Figure 4: The effect of adding bigrams and trigrams for various algorithms. No consistent behaviour can be observed across the datasets. This is contrary to the findings of Wang and Manning (2012) who found that adding bigrams always helped for Topic Classification and Sentiment Analysis. Interestingly while we can reproduce the positive effect of bigrams and trigrams on the *reviews* dataset, we find that bigrams or trigrams do not help on the full *20news* dataset (Wang and Manning (2012) used 3 different 2-class subsets of the *20news* dataset). We hypothesise that the disparity between the findings in Wang and Manning (2012) is due to the different characteristics between the Twitter datasets in our study, and the ones used by in their experiments.

EM-CWF — MNB add-1 but also highlighted that the performance of NB-EM combinations can considerably be improved when their hyperparameters are optimised. We showed that with these modifications NB-EM is competitive with SFE and FM on our datasets. Overall we demonstrated that the modifications to Naïve Bayes and EM, and the usage of alternative semi-supervised learning algorithms, outperformed the baseline configuration on almost all datasets. We furthermore pointed out that none of the semi-supervised learning algorithms we evaluated can consistently make effective use of a large amount of unlabelled data. Lastly, we presented the result that adding bigrams or trigrams does not consistently improve performance in an agile scenario on our datasets.

Acknowledgments

We thank Jeremy Reffin, the TAG lab team and our anonymous reviewers for their helpful comments.

References

Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 492–499, Washington, DC, USA. IEEE Computer Society.

Jaime Bartlett and Richard Norrie. 2015. Immigration on twitter. http://www.demos.co.uk/publications/immigration-on-twitter.

Jaime Bartlett, Jonathan Birdwell, and Louis Reynolds. 2014a. Like, share, vote. http://www.demos.co.uk/publications/likesharevote.

Jaime Bartlett, Richard Norrie, Sofia Patel, Rebekka Rumpel, and Simon Wibberley. 2014b. Misogyny on twitter. http://www.demos.co.uk/publications/misogyny.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8.

Nitesh V. Chawla and Grigoris I. Karakoulas. 2005. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *J. Artif. Intell. Res. (JAIR)*, 23:331–366.

Pedro Domingos and Michael Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130, November.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95)*.

Michael Lucas and Doug Downey. 2013. Scaling semi-supervised naive bayes with feature marginals. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 343–351, Sofia, Bulgaria, August. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Micol Marchetti-Bowick and Nathanael Chambers. 2012. Learning for microblogs with distant supervision: political forecasting with twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 603–612, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press.

Vangelis Metsis, Ion Androutsopoulos, and Paliouras Georgios. 2006. Spam filtering with naive bayes – which naive bayes? In *Third Conference on Email and Anti-Spam (CEAS)*.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, May.

Kevin Dela Rosa, Rushin Shah, Bo Lin, Anatole Gershman, and Robert Frederking. 2011. Topical clustering of tweets. In *Proceedings of the ACM SIGIR Special Interest Group on Information Retrieval's 3rd Workshop on Social Web Search and Mining (SIGIR: SWSM 2011)*. ACM.

Burr Settles and Xiaojin Zhu. 2012. Behavioral factors in interactive training of text classifiers. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 563–567, Stroudsburg, PA, USA. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 2013. Exploiting topic based twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jiang Su, Jelber S. Shirab, and Stan Matwin. 2011. Large scale text classification using semi-supervised multinomial naive bayes. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 97–104, New York, NY, USA. ACM.

A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea, July. Association for Computational Linguistics.

Jason Weston, Sumit Chopra, and Keith Adams. 2014. #tagspace: Semantic embeddings from hashtags. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1822–1827, Doha, Qatar, October. Association for Computational Linguistics.

Simon Wibberley, David Weir, and Jeremy Reffin. 2013. Language technology for agile social media science. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 36–42, Sofia, Bulgaria, August. Association for Computational Linguistics.

Simon Wibberley, David Weir, and Jeremy Reffin. 2014. Method51 for mining insight from social media datasets. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 115–119, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.