A Replication Study of the Top Performing Systems in SemEval Twitter Sentiment Analysis

Efstratios Sygkounas[1], Giuseppe Rizzo[2], Raphaël Troncy[1]

[1] EURECOM, Sophia Antipolis, France,
{sygkouna,troncy}@eurecom.fr
[2] ISMB, Turin, Italy,
giuseppe.rizzo@ismb.it

Abstract. We performed a thorough replicate study of the top performing systems in the yearly SemEval Twitter Sentiment Analysis task. We highlight and discuss differences among the results obtained by those systems that have been officially published and the ones we are able to compute. Learning from the studies being made on the systems, we also propose SentiME, an ensemble system composed of five state-of-the-art sentiment classifiers. SentiME trains the different classifiers using the Bootstrap Aggregating Algorithm. The classification results are then aggregated using a linear function that averages the classification distributions of the different classifiers. SentiME has also been evaluated over the SemEval2015 test set, properly trained with the SemEval2015 train test. We show that SentiME would outperform the best ranked system of the challenge.

1 Introduction

Replication studies are a core element of scientific research. They play a crucial role, either during a peer-review process, or *a posteriori*, for validating results and approaches and enabling further scientific progress. They aim to generate the same overall conclusions rather than producing the same exact figures [3, 4, 2]. We observe a steady rise of challenges organized by particular scientific communities, that aim to share common datasets, tasks and scorers to statistically evaluate results and enable comparison of approaches. There is also a strong encouragement from the research community to publish software source code, scripts and models as citable resources alongside traditional papers describing a particular approach and its evaluation.

In this paper, we propose to perform a thorough replication and reproduction study of the top systems that have competed to the yearly SemEval Twitter Sentiment Analysis tasks [17]. Specifically, we replicated the Webis system [6], an ensemble system of four state-of-art sub-classifiers that ranked first in SemEval 2015 among forty different systems. These four individual sub-classifiers have themselves participated during previous SemEval years where they were also among the top performing systems. The ensemble approach adopted by the Webis system has also inspired us to propose the SentiME system that adds another classifier on top of the Webis system.

Similar to [8], we adopt the following definitions that have been proposed during the recent SIGIR 2015 workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR)[3]:

- **Replicability**: "Repeating a previous result under the original conditions (e.g. same system configuration and datasets)";
- **Reproducibility**: "Reproducing a previous result under different, but comparable conditions";
- **Generalizability**: "Applying an existing, empirically validated technique to a different task/domain than the original one".

We aim to perform a replication study using the Webis source code and the pre-trained models provided by the authors. We also perform a reproducible study using again the Webis system but this time, training ourselves the models using the same features reported by the authors. Finally, our generalizability study leads to the creation of the SentiME system, an ensemble approach developed on top of the Webis system where a fifth classifier, namely the off-the-shelves Stanford Sentiment System, is added. We have also evaluated SentiME on a different dataset composed of one million Amazon reviews of products [21].

The remainder of this paper is structured as follows: in Section 2, we describe the particular SemEval task, datasets and systems we aim to replicate. In Section 3, we present our replicate study, where we first tried to replicate the results of the Webis system using the models provided by the authors, and then, by re-training ourselves those models. We present SentiME, our own ensemble system in Section 4 and we show that it would outperform the best performing systems in SemEval. We provide some lessons learned during this replication study (Section 5) before concluding and outlining future work (Section 6).

2 SemEval Twitter Sentiment Analysis Task

SemEval (Semantic Evaluation) is an ongoing series of evaluations of computational semantic analysis systems, where "semantic analysis" refers to a formal analysis of meaning, and "computational" refers to approaches that in principle support effective implementation [1]. While the series of evaluation has first focused on word sense disambiguation, it has evolved to investigate the interrelationships among the elements in a sentence (e.g., semantic role labeling), relations between sentences (e.g., coreference), and the nature of what we are saying in sentences (semantic relations and sentiment analysis). Since 2012, SemEval is part of the *SEM Conference. In 2013, the organizers have created a new task about Sentiment analysis in Twitter which was identified as the Task 2 in SemEval-2013, the Task 9 in SemEval-2014 and the Task 10 in SemEval-2015.

The general goal of this task is to better understand how the sentiment is expressed in short text messages such as tweets which are still considered as constrained 140 characters message. Each year, the task is actually divided into two sub-tasks: Subtask

[3] `https://sites.google.com/site/sigirrigor/`

A about “Contextual Polarity Disambiguation” and Subtask B about “Message Polarity Classification”. In 2015, the organizers have added three additional sub-tasks: Subtask C about “Topic-Based Message Polarity Classification”, Subtask D about “Detecting Trends Towards a Topic” and Subtask E about “Determining strength of association of Twitter terms with positive sentiment (or, degree of prior polarity)”. This replication study focuses on systems participating and ranked in the top positions for the Subtask B since the inception of the competition.

2.1 SemEval SubtaskB in 2013-2015: task and corpus

The Subtask B asks participants to classify a given sentence (message or tweet) in three possible categories according to the overall sentiment which is conveyed: negative, neutral or positive. For tweets that transmit both positive and negative sentiment, the stronger should be chosen. SemEval provides each participation team a common training and test datasets, which have been annotated by the organizers. This allows different teams to compete on one controlled environment and to compare the performance of different algorithms and approaches in a fair way.

In 2013, the SemEval organizers have collected tweets to compose the training dataset over a one-year period. The test dataset corresponds to a three-months period collection realized before the competition. The raw tweets being heavily skew towards neutral, the SemEval organizers have filtered out large amount of neutral tweets and organize them in topics. Table 1 shows the total number of tweets and their distribution following the classification positive, negative and neutral.

Corpus	**Positive**	**Negative**	**Neutral**	**Total**
Twitter2013-train	3,662	1,466	4,600	9,728
Twitter2013-dev	575	340	739	1,654
Twitter2013-test	1,572	601	1,640	3,813
Twitter2014-test	982	202	669	1,853
Twitter2014-sarcasm	33	40	13	86
LiveJournal2014-test	427	304	411	1,142
Twitter2015-test	1040	365	987	2392

Table 1. Datasets statistics for the Subtask B of SemEval-2015

Annotations have been performed by Amazon Mechanical Turk (Turkers) resulting ultimately in a gold standard file composed of four fields: the original tweet’s ID, the tweet’s gold standard ID, the tweet’s polarity and the textual content of the tweet. These fields are stored into column separated tsv files. Table 2 provides some examples following this schema.

Systems participating in SemEval2015 must generate a output file which contains for each tweet, its classification result: Positive, Neutral or Negative. A system output is then compared with the gold standard file by the organizers who compute the Precision of positive, the Recall of positive, the Precision of negative and the Recall of negative tweets classification. The F-scores for both positive (equation 1) and negative (equation

Tweet ID	**Gold standard ID**	**Polarity**	**Tweet content**
522931511323275264	T15111115	positive	Catch Rainbow Valley at ...
522838326126989314	T15111137	neutral	I wonder if Billy Joe ...
520829332525441024	T15111111	negative	Saturday without Leeds ...

Table 2. SemEval's datasets structure.

2) are computed to finally generate the general F-score (equation 3) which is used to rank the various participating systems.

$$F_{pos} = 2 * \left(\frac{P_{pos} * R_{pos}}{P_{pos} + R_{pos}} \right) \tag{1}$$

$$F_{neg} = 2 * \left(\frac{P_{neg} * R_{neg}}{P_{neg} + R_{neg}} \right) \tag{2}$$

$$F = \frac{P_{pos} + R_{neg}}{2} \tag{3}$$

2.2 Top scoring systems in 2013-2015

NRC-Canada The NRC-Canada team ranked 1st in SemEval 2013, using a SVM classifier to extract the sentiment from tweets [12]. They used different lexicons such as lists of words assigned with either a positive or a negative sentiment, the NRC Emotion Lexicon [13, 14], the MPQA Lexicon [22], and the Bing Liu Lexicon [9]. They also used specific Twitter-based lexicon such as the NRC Hashtag Sentiment Lexicon [11] and the Sentiment140 Lexicon [12]. The system is trained with a linear kernel of a state-of-the-art Support Vector Machine (SVM) algorithm. A pre-processing phase enables to make the tweets easier to be processed. Each tweet is then represented by a feature vector composed of: N-grams, ALLCAPS, POS, Polarity Dictionaries, Punctuation Marks, Emoticons, Word Lengthening, Clusters and Negation.

GU-MLT-LT The GU-MLT-LT team ranked 2nd in SemEval 2013, using a linear classifier trained by stochastic gradient descent with hinge loss and elastic net regularization for their predictions [5]. They also perform a pre-processing phase for the tweets where they included a variety of linguistics and lexical features such as: Normalized Uni-grams, Stems, Clustering, Polarity Dictionary and Negation.

KLUE The KLUE team ranked 5th in SemEval 2013, using a simple bag-of-words models with three different features unigrams, unigrams and bigrams, and an extended unigram model that includes a simple treatment of negation [16]. They also pre-process the tweets and they used features based on a sentiment dictionary such as SentiStrength and an extended version of AFINN-111. Large-vocabulary distributional semantic models (DSM) have been used in order to obtain better word coverage, constructed from a version of the English Wikipedia[4] and the Google Web 1T 5-Grams databases[5]. Finally, they included features based on emoticons and slang abbreviations mostly used on the Web and manually classified by themselves.

TeamX The TeamX team ranked 1st in SemEval 2014, using a variety of pre-processors and features [10]. More specifically, the TeamX system used a large variety of lexicons categorized into "FORMAL" and "INFORMAL" such as AFINN-111 [15], Bing Lius Opinion Lexicon1 [9], General Inquirer [20], MPQA Subjectivity Lexicon [22], NRC Hashtag Sentiment Lexicon [11], Sentiment140 Lexicon Lexicon [12] and SentiWordNetBaccianella2010. Furthermore, they used additional features such as word ngrams, character ngrams, clusters and word senses. Eventually, they fed these features to a supervised machine learning algorithm which utilizes Logistic Regression (LIBLINEAR).

Webis The Webis team ranked 1st in SemEval 2015, using an ensemble method over the four state-of-the-art systems previously described: NRC-Canada, GU-MLT-LT, KLUE and TeamX [6]. Initially, they selected the wining system of SemEval-2013, namely NRC-CANADA, and they manually choose the remaining three systems having as one and only criterion the level of dissimilarity of these systems with respect to NRC-CANADA. Having dissimilar systems in an ensemble is very important since it ultimately leads to a bigger diversity of features and lexicons being used. In other words, each one of the sub-classifiers complement each other which makes the ensemble method special and particularly effective on such a challenge.

In the Webis ensemble system, the authors did not use the classification results of each of the four sub-classifiers but instead, they used their confidence scores. Hence, if two sub-classifiers are not confident enough to provide a classification, the final sentiment will only depend on the other two remaining sub-classifiers providing a higher confidence. The authors also preferred not to build a weighting schema but to use a linear function which averages the classification distributions provided by the four sub-classifiers and produce the final classification according to the maximum value of the labels in the average classification distribution. In summary, the Webis system works as follows: the sub-classifiers are trained individually; the ensemble ignores the individual classification results coming the four sub-classifiers but it considers the confidence scores (possibilities) for each class (positive, neutral and negative). The final classification is done by averaging the confidence scores for each class, the highest confidence score providing the final classification.

2.3 Stanford Sentiment System

The Stanford Sentiment System [19] is one of the sub-systems of the Stanford NLP Core toolkit. It contains the Stanford Tree Parser, a machine-learning model which can

[4] The pre-processed and linguistically annotated Wackypedia corpus they used is from `http://wacky.sslmit.unibo.it/doku.php?id=corpora`

[5] `http://googleresearch.blogspot.fr/2006/08/all-our-n-gram-are-belong-to-you.html`

parse the input text into Stanford Tree format and use some existing models, some of them are trained especially for parsing tweets. The Stanford Sentiment Classifier is at the heart of the system. This classifier takes as input Stanford Trees and outputs the classification results for Stanford Trees. The Stanford Sentiment Classifier provides also useful detailed results such as classification label and classification distribution on all the nodes in the Stanford Tree.

The Stanford Sentiment System is a Recursive Neural Tensor Network trained on the Stanford Sentiment TreeBank that is the first corpus with fully labeled parse trees which makes possible training a model with large and labeled dataset. This model store the information for compositional vector representations, its size of parameters is not very large and the computation cost is reasonable. Moreover, the Stanford Sentiment System can capture the meaning of longer phrases and shows a great strength in classifying negations. It beats the bag of word approaches when predicting fine-grained sentiment labels.

While this system has never participated in previous SemEval years, we have decided to include it as an off-the-shelves classifier in a new ensemble system called SentiME (Section 4). In particular, we will demonstrate that its addition enables to outperform previous system on the sarcasm corpus.

3 Replication Study

3.1 Methodology

The initial goal of a replication study is to identify and to select state-of-the-art approaches that have performed relatively well in a given settings and to compare results with the ones published by the original authors. In this replication study, we first focused our research on the three top performing systems of the SemEval-2015: Webis [6] (1st), UNITN [18] (2nd) and Lsislif [7] (3rd). We decided to replicate the Webis system for three reasons: first, it is the best performing system in SemEval-2015; second, it is a state-of-the-art implementation of an ensemble method combining four sub-classifiers which themselves participated in previous years of SemEval thus giving a broader scope of this replicate study. Furthermore, the ensemble method give us more flexibility to achieve the generalizability we are looking for, in particular when including new classifiers to improve the overall system; third, but not least, the Webis source code has been released openly. While replicating a system without having access to the source code is not impossible, it is much harder, a paper or a technical report being rarely self-sufficient to re-implement a system from scratch. We have contacted the UNITN team, letting them know about our research, and asked if they were willing to share with us their source code but we did not receive any response. Similarly, the Lsislif paper description leaves too many open questions for deciding to re-implement this particular system.

The Webis team has released both the source code of the system and the models they have trained for the SemEval challenge at `https://github.com/webis-de/ECIR-2015-and-SEMEVAL-2015`. We verified that this version corresponds to the system which has being used to report official results at SemEval.

We manage to download and cleanse all the datasets through a download script from the SemEval's official website[6] via the Twitter API. A small number of tweets are not accessible anymore via the API. Consequently, the training and test sets are slightly different, which might have affected the performance of our replicate system. Table 3 reports the differences in terms of downloaded tweets across all datasets compared to the number of tweets SemEval-2015 organizers presented in [17].

Corpus	**Nb tweets collected**	**Nb tweets originally in SemEval**	**Multiple Tweets**
SemEval2013-train+dev-B	11,338	11,382	50
SemEval2013-test-gold-B	3,813	3,813	3
SemEval2014-test-gold-B	1,853	1,853	0
SemEval2014-test-sarcasm-B	86	86	0
SemEval2015-gold-B	2,390	2,392	11
SemEval2015-test-sarcasm-B	60	N/A	0

Table 3. Comparison of datasets: some tweets are unfortunately not available anymore

We observe that we can almost collect all the data that has been officially provided by the SemEval organizers. The main differences, in the SemEval2013-train+dev-B and SemEval2015-gold-B datasets, are either due to the fact that some tweets have been deleted or made inaccessible, or because they are out of the time window that SemEval used to publish its data. Due to the Twitter TOS, nobody can publicly publish the original tweets content. However, the small volume of missing tweets do not hinder the validity of our replicate experiment.

The Multiple Tweets column shows a very interesting phenomenon in the dataset. Each count in Multiple Tweets represents one true identical tweet (exact same tweet ID) appearing more than once in the dataset. This is not due to multiple users publishing the same tweet content (or using the RT functionality) but because SemEval does not filter out some multiple tweets either intentionally or unintentionally. Although we are not sure what was the purpose of the SemEval organizers, we decided to not filter out the multiple tweets because the other participating teams such as Webis did not claim to have performed this operation.

3.2 Replicating Webis using pre-trained and re-trained models

Due to the fact that the Webis system is the ensemble of four sub-classifiers and that each sub-classifier is built using the classifier API of WEKA, the performance of the system is based on some external libraries. Versioning (of software libraries and dependencies) is an important aspect to be considered in any replication study. In this study, we removed the old external libraries which are related to Stanford NLP Core from the Webis system and we added the newest version of Stanford NLP Core libraries.

[6] `http://alt.qcri.org/semeval2015/task10/index.php?id=data-and-tools`

We have created two separate workbenches. The first replicate system use the Webis system as well as the models provided by the authors on their github repository. In the second replicate system, we re-train ourselves the various sub-classifiers composing the Webis system: NRC-CANADA, GU-MLT-LT, KLUE and TEAMX. Our replicate system can train each one of the four sub-classifiers individually and test them individually or all together as an ensemble system using any of the dataset we have. We compared the classification results using either the models provided by Webis or the models we were able to train, using the same Webis ensemble configuration (Table 4).

Dataset	**Claimed in papers [17, 6]**	**Webis's Models**	**Our Models**
Replicate Webis system on test 2013	68.49	69.62	70.06
Replicate Webis system without TeamX on test 2013	N/A	69.04	70.34
Replicate Webis system on test 2014	70.86	66.65	69.31
Replicate Webis system without TeamX on test 2014	N/A	66.51	68.56
Replicate Webis system on test 2015	64.84	66.17	66.57
Replicate Webis system without TeamX on test 2015	N/A	65.58	66.19

Table 4. F-scores for pre-trained and re-trained models in comparison with the results reported in the original papers

Concerning the training workflow, the sub-classifiers are trained individually and one attribute file is generated for storing the model for each one of the sub-classifiers. We also provide one function in our replicate system to train the four sub-classifiers all together. The training of each individual sub-classifier involves four steps. First, a feature extractor processes all tweets in the training dataset to generate feature vectors from the tweets. This step is different among the four sub-classifiers due to the different features each sub-classifier uses. Second, feature vectors are passed to the classifiers. Third, each sub-classifier is trained to generate the parameters of the classifier. The fourth and final step is to store all those information into an attribute file that represents the model.

Regarding the testing process, each sub-classifier is also processing independently the dataset. The first step is to load all the parameters into the feature extractor and the classifier. Then we pre-process the tweets similar to the training process. The next step is to extract the feature vectors from the cleansed tweet texts and to pass them to the classifier.

Each sub-classifier will give a classification result and a the confidence scores for each of the three classes (positive, neutral and negative). When we aggregate the classification results of the four sub-classifiers, we just use a simple linear function which averages the classification distributions of the four sub-classifiers and classify the tweet according to the label which holds the maximal value in the average classification distribution.

After building our replicate system, we launch several experiments to test whether our replicate system achieve exact or similar results than the ones reported by the Webis team in their paper [6]. Consequently, we perform experiments using the individual sub-classifiers and the ensemble system on the SemEval2013-test, SemEval2014-test and SemEval2015-test datatset using either the models provided by the Webis team or the models we re-trained using the SemEval2013-train+dev corpus. Those results are reported in Table 4.

4 SentiME: Generalizing the Webis System

We originally aim to replicate and to reproduce the Webis system in order to see if we can get comparable results. Our investigations have lead us to generalize the system and to propose SentiME, a new ensemble system that add a fifth sentiment classifier to the Webis system, the Stanford Sentiment System that is used as an off-the-shelf classifier[7]. We also propose to use bagging to boost the training of the ensemble system.

The Stanford Sentiment System is a recursive neural tensor network parsed by the Stanford Tree Bank. It is significantly different from all the other classifiers used on tweets polarity prediction and it shows great performance on negative classification. Hence, the negative recall of the sole Stanford Sentiment System is over 90% on average which makes it trustworthy to detect negation (Table 5). We want to investigate whether the addition of this new sub-classifier would improve our Webis replicate system.

Corpus	**Negative Recall**
SemEval2014-test-gold-B	0.9108910891089109
SemEval2015-gold-B	0.8980716253443526

Table 5. Negative recall of the sole Stanford Sentiment System on SemEval datasets

The classification distribution provided by the Stanford Sentiment Classifier consists of five labels: very positive, positive, neutral, negative and very negative. Consequently, we need to map these five labels into the three classes expected by the SemEval challenge for a consistent integration with our replicate system. We only extract the root classification distribution because it represents the classification distribution of the whole tweet text. We have tested different configurations for mapping the Stanford Sentiment System classification to the three classes modem. According to the results of these tests, we decide to use the following mapping algorithm: very positive and positive are mapped to Positive, neutral are mapped to Neutral and negative and very negative are mapped to Negative.

There are multiple ways to do an ensemble of different systems. In the case of SentiME, we propose to use the Stanford Sentiment System as an off-the-shelve classifier

[7] The Stanford Sentiment System will not be trained with the task datasets.

that will not be re-trained. We also propose to use a bagging algorithm for boosting the training of the four other sub-classifiers. To perform bagging, we generate new training data set for each sub-classifier using uniformly sampling with replacement. In order to simulate this procedure, we use the pseudo random number generator which is provided by the Java Util package to generate random tweet indexes used to sample from the loaded tweet list. Afterwards, we provide the system time as seed to Random Class to increase the randomness between different sub-classifiers and we carefully pick and test the size of the bootstrap samples. After training the four sub-classifiers, we store their resulting attribute files for further usage. When we test these models on the test dataset, the aggregating function is just one simple linear function which averages the classification results from the four sub-classifiers as done by Webis.

Due to the fact that bagging introduces some randomness into the training process, and that the size of the bootstrap samples are not fixed, we decide to perform multiple experiments with different sizes ranging from 33% to 175%. We perform the training process three times and get three models to test for each size. We observe that doing bagging with 150% of the initial dataset size leads to the best performance in terms of F1 score (Table 6).

Model	**19,842(175%)**	**17,007(150%)**	**14,173(125%)**	**11,338(100%)**
Model 1	64.76	65.45	65.26	65.05
Model 2	64.65	65.81	64.40	64.15
Model 3	64.50	65.71	64.29	65.25
	9,000(80%)	**7,525(66%))**	**5,644(50%))**	**3,780(33%)**
Model 1	63.37	63.80	64.64	62.81
Model 2	64.54	64.92	62.93	62.85
Model 3	63.54	64.67	63.85	61.65

Table 6. Experiments performed with different bagging sizes on SemEval2013-train+dev-B (11,338 original size) training dataset

We choose the linear averaging function as the aggregating function because this is the simplest method and the performance of the ensemble system is easy to predict. To be more precise, the linear aggregating function averages the classification distributions of each sub-classifiers and choose the polarity which holds the maximum value among the average classification distributions as the final classification.

We already mention the problem of multiple identical tweets in the dataset, where the same tweet ID is present several times with different gold standard ID but also different gold standard polarity. In this experiment, we filter out those multiple tweets to evaluate the SentiME system.

We performed four different experiments to evaluate the performance of SentiME compare to our previous replicate of the Webis system:

1. Webis replicate system: this is the replicate of the Webis system using re-trained models as explained in the Section 3;

2. SentiME system: this is the ensemble system composed of the four sub-classifiers used in by Webis plus the Stanford Sentiment System. The ensemble uses a bagging approach for the training phase;
3. Webis replicate system without TeamX;
4. SentiME system without TeamX;

The experiments 3 and 4 are variations of the experiments 1 and 2 where we simply remove the TeamX sub-classifier, based on the observation that this particular sub-classifier plays a similar role than the Stanford Sentiment system.

The Table 7 reports the F-scores for the four different set ups we described above on four different datasets: two datasets contain regular tweets and two datasets contain sarcasm tweets. We evaluated the four sub-classifiers on SemEval2014 test data set, SemEval2014 sarcasm data set, SemEval2015 test data set and SemEval2015 sarcasm data set in order to figure out whether the Stanford Sentiment System has significant impacts on the performance of our ensemble system. The last row of the Table 7 presents the F scores of the Webis system as reported in the authors' paper [6].

System	**SemEval2014-test**	**SemEval2014-sarcasm**	**SemEval2015-test**	**SemEval2015-sarcasm**
Webis Replicate system	**69.31**	60.00	66.57	54.19
SentiME system	68.27	**62.57**	**67.39**	**60.92**
Webis replicate system without TeamX	68.56	62.04	66.19	56.86
SentiME system without TeamX	69.27	62.04	66.38	58.92
Webis	70.86	49.33	64.84	53.59

Table 7. F-scores for the four systems on four different datasets (highest scores are in bold)

We observe that the SentiME system outperforms the Webis Replicate system on all datasets except on the SemEval2014-test, in which the SentiME system without the TeamX sub-classifier has almost the same performance than the Webis Replicate system. Concerning the performance on both sarcasm datasets, it is clear that that SentiME system improves the F score by respectively 2,5% and 6,5% on SemEval2014-sarcasm and SemEval2015-sarcasm datasets. However, it is unclear why we observe a significant difference of performance on the SemEval2014-sarcasm dataset between the original Webis system (49.33%) and our replicate (60%). The fact that the dataset is extremely small (only 86 tweets) prevents us to draw any conclusion.

We notice that some features used in TeamX come from the Stanford NLP Core package and we assume that TeamX shares some common characteristics with the Stanford Sentiment System. Since the idea of our experiments was to figure out what benefits the Stanford Sentiment System can bring to our replicate system, we consider it is reasonable to exclude the TeamX sub-classifier from our replicate system.

The complete workflow of the SentiME system is depicted in the Figure 4. The SentiME system trains the four sub-classifiers independently and the results are stored into four different attribute files. Then, concerning the test process, each sub-classifier reads the attribute files as well as the test dataset. The Stanford Sentiment System just reads the test dataset. The final step is to average the five different classification results of the sub-classifiers in order to derive the final classification result.

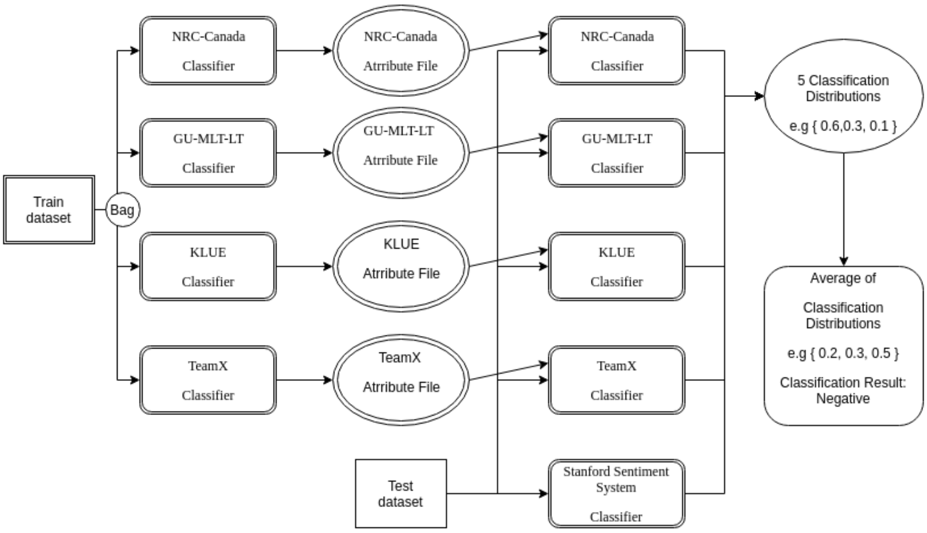

Fig. 1. Workflow of training and test for SentiME

5 Lessons Learned

A few aspects emerged from this comprehensive replication study:

Paper and source code: We started from replicating the experimental setup of the Webis system by studying and implementing the approach reported in the paper. We found the paper not self contained for the settings: we believe that this is partially due do the page limits authors have to respect for complying with publication rules. The source code of the system has helped the replication study presented in this paper. We encounter a few minor issues related to the existence of libraries not included in the source code. We can conservatively state that the availability of the source code has significantly helped to pursue this study and to reproduce the results.

Differences in the performance results due to the training data: The pre-trained models provided by Webis are not exactly the same as the re-trained models we have created from the data at disposal. This is significant for the SemEval2014-test (Table 4) and can be explained by the loss we had in the collection of tweets. We have also to consider that older tweets are more likely to be removed or made inaccessible because of the tweet persistence that is dependent of the Twitter platform. This is how we can explain the under-performance of the SentiME system on the same dataset (Table 7).

Differences in performance results due to the differences in the source code: Another topic of discussion is the differences between the F1 scores claimed in SemEval's 2015 and Webis's papers [17, 6] with the F1 scores of the pre-trained models we computed. These differences indicate that our replicate Webis system is not exactly the same than the original Webis system. This is reasonable because the Webis system contains a lot of libraries which have been updated between the time their experiments and our experiments have been performed. The differences we noticed in the datasets organizers provided for the SemEval competition played also an important role for the final results.

Differences in performance results due to missing features: In addition, another factor that we need to consider about the re-trained models we computed and why they differ is the possibility that the Webis's authors did not detail the full set of features they have used. Feature engineering is an art and the devil is in the details.

6 Conclusion and Future Work

The replication of a prior study is not an easy task since one has always to anticipate possible differences from the original study that may lead to different results. When achieving a thorough replication study, a natural evolution is to propose a generalization for improving a system. The number of tweets we can collect for each dataset is not strictly identical with the numbers reported by the SemEval-2015 organizers [17]. Nevertheless, we manage to replicate the Webis system and to reproduce its functioning by re-training ourselves the models being used. We observe that the Stanford Sentiment System is heavily skew towards negative classification and share a lot of commonalities with the TeamX sub-classifier which is being used by the Webis systems. We also demonstrate that the Stanford Sentiment System improves the performance of a sentiment detection system on a sarcasm dataset.

We manage to improve the Webis system by 1% in the general case by introducing a fifth sub-classifier (the Stanford Sentiment System) and by boosting the training with bagging 150% of the original training dataset while filtering out the multiple tweets. The SentiME system also outperforms the Webis system by 6,5% on the particular and more difficult sarcasm dataset. Additional experiments performed on product reviews confirm that the use of bagging during the training phase is the main driver for improving significantly the performance [21]. The SentiME system is available at `https://github.com/MultimediaSemantics/sentime` and is itself fully replicable.

We suggest, for future work, to improve the aggregating algorithm used in our experiments which, so far, is a simple linear function which averages the classification distributions of each sub-classifier. This is the most basic aggregating function and there is consequently a lot of space for improvement. We suggest to use some weighted aggregating functions and perform some related experiments in order to find out the best possible system set up. Moreover, it is worth trying to train the Stanford Sentiment System with the SemEval training datasets. This will require to convert the training dataset into the Stanford Tree Bank format.

Concerning the training process, finding the best size of bootstrap samples is a real challenge. One should not only look at the performance improvement in terms of the F scores, but should also consider whether a stable training process can be established since bagging introduces some randomness to the training process. Consequently, a series of very fine-grained experiments that may take a long time to run must be performed. The aggregating algorithm we use in our bagging process is the linear function which averages the classification distributions of each sub-classifier. Since this linear function is very simple and does not involve any careful consideration, it is possible that the performance of the system could be improved by replacing the aggregating algorithm by a new technique.

Acknowledgments

This work was partially supported by the innovation activity 3cixty (14523) of EIT Digital and by the European Union's H2020 Framework Programme via the FREME Project (644771).

References

1. Blackburn, P., Bos, J.: Representation and Inference for Natural Language: A First Course in Computational Semantics. CSLI (2005)
2. Buchert, T., Nussbaum, L.: Leveraging business workflows in distributed systems research for the orchestration of reproducible and scalable experiments. In: 9th French conference on MAnifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (2012)
3. Dalle, O.: On reproducibility and traceability of simulations. In: WSC - Winter Simulation Conference (2012)
4. Drummond, C.: Replicability is not Reproducibility: Nor is it Good Science. In: Workshop on Evaluation Methods for Machine Learning (2009)
5. Günther, T., Furrer, L.: GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent. In: 7th International Workshop on Semantic Evaluation (SemEval-2013) (2013)
6. Hagen, M., Potthast, M., Büchner, M., Stein, B.: Webis: An Ensemble for Twitter Sentiment Detection. In: 9th International Workshop on Semantic Evaluation (SemEval-2015) (2015)
7. Hamdan, H., Bellot, P., Bechet, F.: Lsislif: Feature Extraction and Label Weighting for Sentiment Analysis in Twitter. In: 9th International Workshop on Semantic Evaluation (SemEval-2015) (2015)
8. Hasibi, F., Balog, K., Bratsberg, S.E.: On the Reproducibility of the TAGME Entity Linking System. In: 38th European Conference on Information Retrieval (ECIR) (2016)
9. Hu, M., Liu, B.: Mining Opinion Features in Customer Reviews. In: 19th National Conference on Artificial Intellgience (2004)
10. Miura, Y., Sakaki, S., Hattori, K., Ohkuma, T.: TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data. In: 8th International Workshop on Semantic Evaluation (SemEval-2014) (2014)

11. Mohammad, S., Kiritchenko, S.: Using Hashtags to Capture Fine Emotion Categories from Tweets. In: Special issue on Semantic Analysis in Social Media, Computational Intelligence (2012)
12. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In: 7th International Workshop on Semantic Evaluation (SemEval-2013) (2013)
13. Mohammad, S., Turney, P.: Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In: NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (2010)
14. Mohammad, S., Turney, P.: Crowdsourcing a WordEmotion Association Lexicon. In: Computational Intelligence (2013)
15. Nielsen, F.: AFINN. Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark (2011), `http://www2.imm.dtu.dk/pubdb/p.php?6010`
16. Proisl, T., Greiner, P., Evert, S., Kabashi, B.: KLUE: Simple and robust methods for polarity classification. In: 7th International Workshop on Semantic Evaluation (SemEval-2013) (2013)
17. Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., Stoyanovm, V.: SemEval-2015 Task 10: Sentiment Analysis in Twitter. In: 9th International Workshop on Semantic Evaluation (SemEval-2015) (2015)
18. Severyn, A., Moschitti, A.: UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In: 9th International Workshop on Semantic Evaluation (SemEval-2015) (2015)
19. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C., Ng, A., Potts, C.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2013)
20. Stone, P., Dunphy, D., Smith, S.: The General Inquirer: A Computer Approach to Content Analysis. Journal of Regional Science 8(1), 113116 (1968)
21. Sygkounas, E., Rizzo, G., Troncy, R.: Sentiment Polarity Detection From Amazon Reviews: An Experimental Study. In: Semantic Web Evaluation Challenges (2016)
22. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (2005)