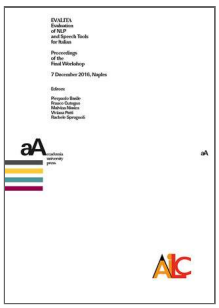

Pierpaolo Basile, Franco Cutugno, Malvina Nissim, Viviana Patti and Rachele Sprugnoli (dir.)

EVALITA. Evaluation of NLP and Speech Tools for Italian
Proceedings of the Final Workshop 7 December 2016, Naples

Accademia University Press

On the performance of B4MSA on SENTIPOLC'16

Daniela Moctezuma, Eric S. Tellez, Mario Graff and Sabino Miranda-Jiménez

DOI: 10.4000/books.aaccademia.2017
Publisher: Accademia University Press
Place of publication: Torino
Year of publication: 2016
Published on OpenEdition Books: 28 August 2017
Serie: Collana dell'Associazione Italiana di Linguistica Computazionale
Electronic ISBN: 9788899982553

On the performance of B4MSA on SENTIPOLC'16

Daniela Moctezuma
CONACyT-CentroGEO
Circuito Tecnopolo Norte No. 117,
Col. Tecnopolo Pocitos II, C.P. 20313, Ags, México
dmoctezuma@centrogeo.edu.mx

Eric S. Tellez
Mario Graff
Sabino Miranda-Jiménez
CONACyT-INFOTEC
Circuito Tecnopolo Sur
No 112, Fracc. Tecnopolo Pocitos II,
Ags, 20313, México.
eric.tellez@infotec.mx
mario.graff@infotec.mx
sabino.miranda@infotec.mx

Abstract

This document describes the participation of the INGEOTEC team in SENTIPOLC 2016 contest. In this participation two approaches are presented, B4MSA and B4MSA + EvoDAG, tested in Task 1: Subjectivity classification and Task 2: Polarity classification. In case of polarity classification, one constrained and unconstrained runs were conducted. In subjectivity classification only a constrained run was done. In our methodology we explored a set of techniques as lemmatization, stemming, entity removal, character-based q-grams, word-based n-grams, among others, to prepare different text representations, in this case, applied to the Italian language. The results show the official competition measures and other well-known performance measures such as macro and micro F1 scores.

Italiano. *Questo documento descrive la partecipazione del team INGEOTEC alla competizione SENTIPOLC 2016. In questo contributo sono presentati due approcci, B4MSA e B4MSA + EvoDAG, applicati al Task 1: Subjectivity classification e Task 2: Polarity classification. Nel caso della classificazione della polarit, sono stati sottomessi un run constrained ed un run unconstrained. Per la classificazione della soggettivita, stato sottomesso solo un run constrained. La nostra metodologia esplora un insieme di tecniche come lemmatizzazione, stemming, rimozione di entit, q-grammi di caratteri, n-grammi di parole, ed altri, al fine di ottenere diverse rappresentazioni del testo. In questo caso essa applicata alla lingua italiana. I risultati qui presentati sono due: le metriche della competizione ufficiale ed altre misure note della performance, come macro F1 e micro F1.*

1 Introduction

Nowadays, the sentiment analysis task has become a problem of interest for governments, companies, and institutions due to the possibility of sensing massively the mood of the people using social networks in order to take advantage in decision-making process. This new way to know *what are people thinking* about something imposes challenges to the natural language processing and machine learning areas, the first of all, is that people using social networks are kindly ignoring formal writing. For example, a typical Twitter user do not follow formal writing rules and introduces new lexical variations indiscriminately, the use of emoticons and the mix of languages is also the common lingo. These characteristics produce high dimensional representations, where the curse of dimension makes hard to learn from examples.

There exists a number of strategies to cope with the sentiment analysis on Twitter messages, some of them are based on the fact that the core problem is fixed: we are looking for evidence of some sentiment in the text. Under this scheme a number of dictionaries have been described by psychologists, other resources like SentiWordNet have been created adapting well known linguistic resources and machine learning. There is a lot of work around this approach; however, all these knowledge is language dependent and must exists a deep understanding of the language being analyzed. Our ap-

proach is mostly independent of this kind of external resources while focus on tackling the misspellings and other common errors in the text.

In this manuscript we detail our approach to sentiment analysis from a language agnostic perspective, e.g., no one in our team knows Italian language. We neither use external knowledge nor specialized parsers. Our aim is to create a solid baseline from a multilingual perspective, that can be used as a real baseline for challenges like SENTIPOLC'16 and as a basic initial approximation for sentiment analysis systems.

The rest of the paper is organized in the following sections. Section 2 describes our approach. Section 3 describes our experimental results, and finally Section 4 concludes.

2 Our participation

This participation is based on two approaches. First, B4MSA method, a simple approach which starts by applying text-transformations to the tweets, then transformed tweets are represented in a vector space model, and finally, a Support Vector Machine (with linear kernel) is used as the classifier. Second, B4MSA + EvoDAG, a combination of this simple approach with a Genetic programming scheme.

2.1 Text modeling with B4MSA

B4MSA is a system for multilingual polarity classification that can serve as a baseline as well as a framework to build sophisticated sentiment analysis systems due to its simplicity. The source code of B4MSA can be downloaded freely[1].

We used our previous work, B4MSA, to tackle the SENTIPOLC challenge. Our approach learns based on training examples, avoiding any digested knowledge as dictionaries or ontologies. This scheme allows us to address the problem without caring about the particular language being tackled.

The dataset is converted to a vector space using a standard procedure: the text is normalized, tokenized and weighted. The weighting process is fixed to be performed by TFIDF (Baeza-Yates and Ribeiro-Neto, 2011). After that process, a linear SVM (Support Vector Machines) is trained using 10-fold cross-validation (Burges, 1998). At the end, this classifier is applied to the test set to obtain the final prediction.

[1] https://github.com/INGEOTEC/b4msa

At a glance, our goal is to find the best performing normalization and tokenization pipelines. We state the modeling as a combinatorial optimization problem; then, given a performance measure, we try to find the best performing configuration among a large parameter space.

The list of transformations and tokenizers are listed below. All the text transformations considered are either simple to implement, or there is an open-source library (e.g. (Bird et al., 2009; Řehůřek and Sojka, 2010)) that implement it.

2.2 Set of Features

In order to find the best performing configuration, we used two sort of features that we consider them as parameters: cross-language and language-dependent features.

Cross-language Features could be applied in most similar languages and similar surface features. Removing or keeping *punctuation* (question marks, periods, etc.) and *diacritics* from the original source; applying or not applying the processes of *case sensitivity* (text into lowercase) and *symbol reduction* (repeated symbols into one occurrence of the symbol). *Word-based n-grams (n-words) Feature* are word sequences of words according to the window size defined. To compute the N-words, the text is tokenized and combined the tokens. For example, 1-words (unigrams) are each word alone, and its 2-words (bigrams) set are the sequences of two words, and so on (Jurafsky and Martin, 2009). *Character-based q-grams (q-grams)* are sequences of characters. For example, 1-grams are the symbols alone, 3-grams are sequences of three symbols, generally, given text of size m characters, we obtain a set with at most $m - q + 1$ elements (Navarro and Raffinot, 2002). Finally, *Emoticon (emo) feature* consists in keeping, removing, or grouping the emotions that appear in the text; popular emoticons were hand classified (positive, negative or neutral), included text emoticons and the set of unicode emoticons (Unicode, 2016).

Language Dependent Features. We considered three language dependent features: stopwords, stemming, and negation. These processes are applied or not applied to the text. *Stopwords* and stemming processes use data and the Snowball Stemmer for Italian, respectively, from NLTK Python package (Bird et al., 2009). *Negation* feature markers could change the polarity of the mes-

sage. We used a set of language dependent rules for common negation structures to attached the negation clue to the nearest word, similar to the approach used in (Sidorov et al., 2013).

2.3 Model Selection

The model selection, sometimes called hyper-parameter optimization, is the key of our approach. The default search space of B4MSA contains more than 331 thousand configurations when limited to multilingual and language independent parameters; while the search space reaches close to 4 million configurations when we add our three language-dependent parameters. Depending on the size of the training set, each configuration needs several minutes on a commodity server to be evaluated; thus, an exhaustive exploration of the parameter space can be quite expensive that makes the approach useless.

To reduce the selection time, we perform a stochastic search with two algorithms, *random search* and *hill climbing*. Firstly, we apply random search (Bergstra and Bengio, 2012) that consists on randomly sampling the parameter space and select the best configuration among the sample. The second algorithm consists on a *hill climbing* (Burke et al., 2005; Battiti et al., 2008) implemented with memory to avoid testing a configuration twice. The main idea behind hill climbing is to take a pivoting configuration (in our case we start using the best one found by random search), explore the configuration's neighborhood, and greedily moving to the best neighbor. The process is repeated until no improvement is possible. The configuration neighborhood is defined as the set of configurations such that these differ in just one parameter's value.

Finally, the performance of the final configuration is obtained applying the above procedure and cross-validation over the training data.

2.4 B4MSA + EvoDAG

In the polarity task besides submitting B4MSA which is a constrained approach, we decided to generate an unconstrained submission by performing the following approach. The idea is to provide an additional dataset that it is automatically label with positive and negative polarity using the Distant Supervision approach (Snow et al., 2005; Morgan et al., 2004).

We start collecting tweets (using Twitter stream) written in Italian. In total, we collect more than $10,000,000$ tweets. From these tweets, we kept only those that were consistent with the emoticon's polarity used, e.g., the tweet only contains consistently emoticons with positive polarity. Then, the polarity of the whole tweet was set to the polarity of the emoticons, and we only used positive and negative polarities. Furthermore, we decided to balance the set, and then we remove a lot of positive tweets. At the end, this external dataset contains $4,550,000$ tweets, half of them are positive and the another half are negative.

Once this external dataset was created, we decided to split it in batches of $50,000$ tweets half of them positive and the other half negative. This decision was taken in order to optimize the time needed to train a SVM and also around this number the Macro F1 metric is closed to its maximum value. That is, this number of tweets gives a good trade-off between time needed and classifier performance. In total there are 91 batches.

For each batch, we train a SVM at the end of this process we have 91 predictions (it is use the decision function). Besides these 91 predictions, it is also predicted (using as well the decision function) each tweet with B4MSA. That is, at the end of this process we have 94 values for each tweet. That is, we have a matrix with $7,410$ rows and 94 columns for the training set and of $3,000$ rows and 94 columns for the test set. Moreover, for matrix of the training set, we also know the class for each row. It is important to note that all the values of these matrix are predicted, for example, in B4MSA case, we used a 10-fold cross-validation in the training set in order to have predicted values.

Clearly, at this point, the problem is how to make a final prediction; however, we had built a classification problem using the decision functions and the classes provided by the competition. Thus, it is straight forward to tackle this classification problem using EvoDAG (Evolving Directed Acyclic Graph)[2] (Graff et al., 2017) which is a Genetic Programming classifier that uses semantic crossover operators based on orthogonal projections in the phenotype space. In a nutshell, EvoDAG was used to ensemble the outputs of the 91 SVM trained with the dataset automatically labeled and B4MSA's decision functions.

[2] https://github.com/mgraffg/EvoDAG

3 Results and Discussion

This Section presents the results of the INGEOTEC team. In this participation we did two runs, a constrained and an unconstrained run with B4MSA system, and only a constrained run with B4MSA + EvoDAG. The constrained run was conducted only with the dataset provided by SENTIPOLC'16 competition. For more technical details from the database and the competition in general see (Barbieri et al., 2016).

The unconstrained run was developed with an additional dataset of $4,550,000$ of tweets labeled with Distant Supervision approach. The Distant Supervision is an extension of the paradigm used in (Snow et al., 2005) and nearest to the use of weakly labeled data in (Morgan et al., 2004). In this case, we consider the emoticons as key for automatic labeling. Hence, a tweet with a high level of positive emoticons is labeled as positive class and a tweet with a clear presence of negative emoticons is labeled as negative class. This give us a bigger amount of samples for the dataset for training.

For the constrained run we participate in two task: subjectivity and polarity classification. In the unconstrained run we only participate in polarity classification task. Table 1 shows the results of subjectivity classification Task (B4MSA method), here, $\mathbf{Prec_0}$ is the $Precision_0$ value, $\mathbf{Rec_0}$ is the $Recall_0$ value, $\mathbf{FSc_0}$ is $F-Score_0$ value and $\mathbf{Prec_1}$, $\mathbf{Rec_1}$ and $\mathbf{FSc_1}$ the same for $F-Score_1$ values and $\mathbf{FSc_{avg}}$ is the average value from all F-Scores. The explanation of evaluation measures can be seen in (Barbieri et al., 2016).

Table 2, shows the results on the polarity classification task. In this task our B4MSA method achieves an average F-Score of 0.6054 and our combination of B4MSA + EvoDAG reaches an 0.6075 of average F-Score. These results place us on position 18 (unconstrained run) and 19 (constrained run) of a total of 26 entries.

It is important to mention that the difference between our two approaches is very small; however, B4MSA + EvoDAG is computationally more expensive, so we expected to have a considerable improvement in performance. It is evident that these results should be investigated further, and, our first impression are that our Distant supervision approach should be finely tune, that is, it is needed to verify the polarity of the emoticons and the complexity of the tweets.

Finally, Table 3 presents the measures employed by our internal measurement, that is Macro F1 and Micro F1 (for more details see (Sebastiani, 2002)). These values are from polarity unconstrained run (B4MSA + EvoDAG), polarity constrained run (B4MSA), subjectivity constrained run (B4MSA) and irony classification (B4MSA). We do not participate in irony classification task but we want to show the obtained result from our B4MSA approach on this task.

4 Conclusions

In this work we describe the INGEOTEC team participation in SENTIPOLC'16 contest. Two approaches were used, first, B4MSA method which combine several text transformations to the tweets. Secondly, B4MSA + EvoDAG, which combine the B4MSA method with a genetic programming approach. In subjectivity classification task, the obtained results place us in seventh of a total of 21 places. In polarity classification task, our results place us 18 and 19 places of a total of 26. Since our approach is simple and easy to implement, we take these results important considering that we do not use affective lexicons or another complex linguistic resource. Moreover, our B4MSA approach was tested internally in irony classification task with a result of 0.4687 of macro f1, and 0.8825 of micro f1.

References

Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 2011. *Modern Information Retrieval*. Addison-Wesley, 2nd edition.

Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. 2016. Overview of the EVALITA 2016 SENTiment POLarity Classification Task. In Pierpaolo Basile, Franco Cutugno, Malvina Nissim, Viviana Patti, and Rachele Sprugnoli, editors, *Proceedings of the 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2016)*. aAcademia University Press.

Roberto Battiti, Mauro Brunato, and Franco Mascia. 2008. *Reactive search and intelligent optimization*, volume 45. Springer Science & Business Media.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

$\mathbf{Prec_0}$	$\mathbf{Rec_0}$	$\mathbf{FSc_0}$	$\mathbf{Prec_1}$	$\mathbf{Rec_1}$	$\mathbf{FSc_1}$	$\mathbf{FSc_{avg}}$
0.56	0.80	0.66	0.86	0.67	0.75	0.70

Table 1: Results on Subjectivity Classification

$\mathbf{FScore_{pos}}$	$\mathbf{FScore_{neg}}$	**Combined FScore**
Constrained run (B4MSA)		
0.6414	0.5694	0.6054
Unconstrained run (B4MSA + EvoDAG)		
0.5944	0.6205	0.6075

Table 2: Results on Polarity Classification

Run	Macro F1	Micro F1
Polarity Unconstrained	0.5078	0.5395
Polarity Constrained	0.5075	0.5760
Subjectivity Constrained	0.7137	0.721
Irony Constrained	0.4687	0.8825

Table 3: Micro F1 and Macro F1 results from our approaches

Christopher J.C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Edmund K Burke, Graham Kendall, et al. 2005. *Search methodologies*. Springer.

Mario Graff, Eric S. Tellez, Hugo Jair Escalante, and Sabino Miranda-Jimnez. 2017. Semantic Genetic Programming for Sentiment Analysis. In Oliver Schtze, Leonardo Trujillo, Pierrick Legrand, and Yazmin Maldonado, editors, *NEO 2015*, number 663 in Studies in Computational Intelligence, pages 43–65. Springer International Publishing. DOI: 10.1007/978-3-319-44003-3_2.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37(6):396 – 410. Named Entity Recognition in Biomedicine.

G. Navarro and M. Raffinot. 2002. *Flexible Pattern Matching in Strings – Practical on-line search algorithms for texts and biological sequences*. Cambridge University Press. ISBN 0-521-81307-7. 280 pages.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March.

Grigori Sidorov, Sabino Miranda-Jiménez, Francisco Viveros-Jiménez, Alexander Gelbukh, Noé Castro-Sánchez, Francisco Velásquez, Ismael Díaz-Rangel, Sergio Suárez-Guerra, Alejandro Treviño, and Juan Gordon. 2013. Empirical study of machine learning based approach for opinion mining in tweets. In *Proceedings of the 11th Mexican International Conference on Advances in Artificial Intelligence - Volume Part I*, MICAI'12, pages 1–14, Berlin, Heidelberg. Springer-Verlag.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press.

Unicode. 2016. Unicode emoji chart. `http://unicode.org/emoji/charts/full-emoji-list.html`. Accessed 20-May-2016.