Ingeotec at Rest-Mex: Bag-of-Words Classifiers

Mario Graff[1,2,*], Daniela Moctezuma[3], Eric Tellez[1,2,4] and Sabino Miranda[1,2]

[1]INFOTEC Centro Público de Investigación en Tecnologías de la Información y Comunicación, 112 Circuito Tecnopolo Sur, Parque Industrial Tecnopolo 2, Aguascalientes, 20326, México.

[2]CONAHCYT Consejo Nacional de Humanidades, Ciencia y Tecnología,1582 Insurgentes Sur 1582, Crédito Constructor, Ciudad de México, 03940 México

[3]CentroGEO Centro de Investigación en Ciencias de Información Geoespacial, Circuito Tecnopolo Norte No. 117, Col. Tecnopolo Pocitos II, C.P., Aguascalientes, Ags 20313 México

[4]CICESE Centro de Investigación Científica y de Educación Superior de Ensenada. Carretera Ensenada - Tijuana No. 3918, Zona Playitas, CP. 22860, Ensenada, B.C. México.

Abstract

Sentiment analysis is a powerful tool that can assist businesses and governments in understanding people's opinions and emotions about various topics. Organizations can identify trends, take ideas to improve products and services, improve policies, and enhance the overall experience by analyzing sentiments. Particularly in the tourism industry, sentiment analysis can track customer satisfaction and improve marketing campaigns. This manuscript describes the INGEOTEC system solution to the Rest-Mex challenge at Iberlef 2023, based on the EvoMSA, a multilingual sentiment analysis framework utilizing stacked generalization. The solution uses internal models based on lexical and semantic features, which can be competitive for classification tasks compared with more complex and computationally high-consuming deep learning approaches.

Keywords

Opinion mining in tourism, Bag of Words, Dense Bag of Words,

1. Introduction

Sentiment Analysis is a field of Natural Language Processing that deals with identifying and extracting opinions and emotions from the text. Recently, businesses and organizations have sought to understand how customers feel about their products, services, and brands. Governments are also interested in people's opinions and emotions about public policies, events, and other essential matters; this is where sentiment analysis comes in handy. Businesses and governments can better understand customers' or citizens' needs and want by analyzing people's sentiments toward specific topics. This information can then be used to improve products

IberLEF 2023, September 2023, Jaén, Spain

*Corresponding author.

mario.graff@infotec.mx (M. Graff); dmoctezuma@centrogeo.edu.mx (D. Moctezuma); eric.tellez@infotec.mx (E. Tellez); sabino.miranda@infotec.mx (S. Miranda)

https://mgraffg.github.io/ (M. Graff); https://www.centrogeo.org.mx/areas-profile/dmoctezuma (D. Moctezuma); https://sadit.github.io/ (E. Tellez); https://www.infotec.mx/es_mx/Infotec/sabino-miranda-jimenez (S. Miranda)

0000-0001-6573-4142 (M. Graff); 0000-0003-3038-4642 (D. Moctezuma); 0000-0001-5804-9868 (E. Tellez); 0000-0002-0595-2401 (S. Miranda)

and services, make better policies, and ultimately enhance the overall experience for everyone involved.

In the tourism industry, Sentiment Analysis can track customer satisfaction, identify trends, and improve marketing campaigns [1, 2]. For example, a tourism business could use Sentiment Analysis to identify which aspects of its products and services are most popular with customers and which areas could be improved or to track the popularity of different destinations, activities, and attractions [3, 4, 5].

This manuscript describes our system solution to the Rest-Mex challenge at Iberlef 2023 [6]. Our solution is based on the EvoMSA framework [7], a multilingual sentiment analysis framework built on the concept of stacked generalization [8], an effective way to combine the output of multiple models into a single prediction. In this solution, the best models are based only on the concept of lexical features, i.e., a bag of words, which indicates competitiveness for classification tasks against more complex deep learning approaches.

This manuscript is organized as follows, Section 1.1 presents the introduction and work related to Sentiment Analysis applied to the tourism sector. Section 2 details our proposed system sent to the Rest-Mex challenge. Section 3 presents our results, and some conclusions are given in Section 4. Finally, the appendix has the EvoMSA code to use the models developed.

1.1. Related work

Recent deep learning models are almost based on the Transformer architecture, see [9]. The Transformer architecture is based on the attention mechanism, which allows the model to learn long-range dependencies between words in a sentence. While this can show significant improvements compared with other approaches, it is also computationally expensive. Also, the required resources to train a model are beyond what organizations and research groups can afford. In this sense, a potential user can solve a task by fine-tuning a pre-trained model. Still, full training is prohibitive for most users. Therefore, languages without attention from large companies or research groups will also lack the pre-trained models necessary to solve tasks. Additionally, any modification should be made from these large entities causing users to become highly dependent on what they do or decide to share.

In Sentiment Analysis, there exist two major adopted architectures, Bidirectional Encoder Representations from Transformers known as BERT (see [10]), and the Generative Pre-trained Transformers known as GPT (see [11]).

Clearly, the use of Sentiment Analysis models is very important in the Tourism domain [12], where several approaches have been explored and applied. For instance, BERT models have been applied, in [13], the authors use pre-trained and fine-tuning BETO (BERT model trained on Spanish corpus) and RoBERTuito (pre-trained language model for user-generated content in Spanish) models over the user's opinions to identify the type of attraction and predict the polarity. In [14], the authors executed a fine-tuning process using the BETO model to predict the classes in the specific domain.

The BERT model is used in [15] to analyze sentiments expressed by tourists in China through the Trip.com platform. Another approach, proposed in [16], analyzes travel reviews' sentiment using a Glove-BiLSTM-CMM model and BERT-BiLSTM-CNN models. Ray et al. in [17] introduce a hotel recommendation system through Sentiment Analysis using as input the hotel reviews;

it also uses aspect-based categorization of the reviews. This recommendation system was built using the BERT model, also other text representations to feed a random forest classifier.

2. System description

Sentiment Analysis on tourism can be tackled as text classification problems - in fact, many of the tasks encountered at IberLEF2023 [18] can be posed as text categorization problems. Text classification is a Natural Language Processing task focused on identifying a text's category. A standard approach to tackle text classification problems is to pose it as a supervised learning problem. In supervised learning, everything starts with a dataset composed of pairs of inputs and outputs; in this case, the inputs are texts, and the outputs correspond to the associated labels or categories. The aim is that the developed algorithm can automatically assign a label to any given text independently, whether it was in the original dataset. The feasible categories are only those found on the original dataset. In some circumstances, the method can also inform the confidence it has in its prediction so the user can decide whether to use or discard it.

Following a supervised learning approach requires that the input is in amenable representation for the learning algorithm; usually, this could be a vector. One of the most common methods to represent a text into a vector is to use a Bag of Words (BoW) model, which works by having a fixed vocabulary where each component represents an element in the vocabulary and the presence of it in the text is given by a non-zero value.

The core idea of a BoW is that after the text is normalized and tokenized, each token t is associated with a vector $\mathbf{v_t} \in \mathbb{R}^d$ where the i-th component, i.e., $\mathbf{v_t}_i$, contains the Inverse-Document-Frequency (IDF) value of the token t and $\forall_{j \neq i} \mathbf{v_t}_j = 0$. The set of vectors $\mathbf{v}$ corresponds to the vocabulary, there are d different tokens in the vocabulary, and by definition $\forall_{i \neq j} \mathbf{v_i} \cdot \mathbf{v_j} = 0$, where $\mathbf{v_i} \in \mathbb{R}^d$, $\mathbf{v_j} \in \mathbb{R}^d$, and $(\cdot)$ is the dot product. It is worth mentioning that any token outside the vocabulary is discarded.

Using this notation, a text x is represented by the sequence of its tokens, i.e., $(t_1, t_2, \ldots)$; the sequence can have repeated tokens, e.g., $t_j = t_k$. Then each token is associated with its respective vector $\mathbf{v}$ (keeping the repetitions), i.e., $(\mathbf{v_{t_1}}, \mathbf{v_{t_2}}, \ldots)$. Finally, the text x is represented as:

$$\mathbf{x} = \frac{\sum_t \mathbf{v_t}}{\|\sum_t \mathbf{v_t}\|}, \tag{1}$$

where the sum goes for all the elements of the sequence, $\mathbf{x} \in \mathbb{R}^d$, and $\|\mathbf{w}\|$ is the Euclidean norm of vector $\mathbf{w}$. The term frequency is implicitly computed in the sum because the process allows token repetitions.

The second representation developed relies on using a dense representation based on BoW. The idea is to represent a text in a vector space where the components have a more complex meaning than the BoW model. In BoW, each component's meaning corresponds to the associated token, and the IDF value gives its importance. The complex behavior comes from associating each component to the decision value of a text classifier (e.g., BoW) trained on a labeled dataset that is different from the task at hand, albeit nothing forbids it to be related to it. The datasets from which these decision functions come can be built using a self-supervised approach or annotating texts.

Without loss of generality, it is assumed that there are M labeled datasets, each one contains a binary text classification problem; noting that if a dataset has K labels, then this dataset can be represented as K binary classification problems following the one versus the rest approach, i.e., it is transformed to K datasets.

For each of these M binary text classification problems, a BoW classifier is built using the default parameters (a pre-trained BoW representation and a linear Support Vector Machine (SVM) as the classifier). Consequently, there are M binary text classifiers, i.e., $(c_1, c_2, \ldots, c_M)$. Additionally, the decision function of c_i is a value where the sign indicates the class. The text representation is the vector obtained by concatenating the decision functions of the M classifiers and then normalizing the vector to have length 1.

A text x is represented with vector $\mathbf{x}^{'} \in \mathbb{R}^M$ where the value $\mathbf{x}^{'}_i$ corresponds to the decision function of c_i. Given that the classifier c_i is a linear SVM, the decision function corresponds to the dot product between the input vector and the weight vector $\mathbf{w}_i$ plus the bias $\mathbf{w}_{i_0}$, where the weight vector and the bias are the parameters of the classifier. That is, the value $\mathbf{x}^{'}_i$ corresponds to

$$\mathbf{x}^{'}_i = \mathbf{w}_i \cdot \frac{\sum_t \mathbf{v_t}}{\|\sum_k \mathbf{v_k}\|} + \mathbf{w}_{i_0}, \tag{2}$$

where $\mathbf{v_t}$ is the IDF vector associated to the token t of the text x. In matrix notation, vector $\mathbf{x}'$ is

$$\mathbf{x}^{'} = \mathbf{W} \cdot \frac{\sum_t \mathbf{v_t}}{\|\sum_k \mathbf{v_k}\|} + \mathbf{w_0}, \tag{3}$$

where matrix $\mathbf{W} \in \mathbb{R}^{M \times d}$ contains the weights, and $\mathbf{w_0} \in \mathbb{R}^M$ is the bias. Another way to see the previous formulation is by defining a vector $\mathbf{u_t} = \frac{1}{\|\sum_k \mathbf{v_k}\|}\mathbf{W}\mathbf{v_t}$. Consequently, $\mathbf{x}'$ is defined as:

$$\mathbf{x}' = \sum_t \mathbf{u_t} + \mathbf{w_0}, \tag{4}$$

vectors $\mathbf{u} \in \mathbb{R}^M$ correspond to the tokens; this is the reason we refer to this model as a dense BoW. Finally, the vector representing the text x is the normalized $\mathbf{x}^{'}$, i.e., $\mathbf{x} = \frac{\mathbf{x}^{'}}{\|\mathbf{x}^{'}\|}$.

2.1. BoW parameters

Different BoW representations were created and implemented following the approach described in [19]. The first step was to set all the characters to lowercase and remove diacritics and punctuation symbols. Additionally, the users and the URLs were removed from the text. Once the text is normalized, it is split into bigrams, words, and q-grams of characters with $q = \{2, 3, 4\}$.

The pre-trained BoW is estimated from 4,194,304 (2^{22}) tweets randomly selected. The IDF values were estimated from the collection, and some tokens were selected from the ones found in the collection. Two procedures were used to select the tokens; the first corresponds to selecting the d tokens with the highest frequency, and the other to normalize the frequency w.r.t. their type, i.e., bigrams, words, and q-grams of characters. Once the frequency is normalized, one selects the d tokens with the highest normalized frequency. The value of d is 2^{17}; however, one can also find in the library models for $2^{13}, 2^{14}, \ldots, 2^{17}$.

It is also possible to train the BoW model using the training set; in this case, we used the default parameters. The only difference is that vocabulary size d is unconstrained, i.e., containing all the tokens as found in the training set.

2.2. Dense representation parameters

The dense representations start by defining the labeled datasets used to create them. These datasets are organized in three groups. The first one is composed of human-annotated datasets; we refer to them as *dataset*. The second group contains a set of self-supervised datasets where the objective is to predict the presence of an emoji (these models are referred to as *emoji*). The final group is also a set of self-supervised datasets where the task is to predict the presence of a particular word, namely *keyword*.

Following an equivalent approach used in the development of the pre-trained BoW, different dense representations were created; these correspond to varying the size of the vocabulary and the two procedures used to select the tokens. The vector space create by the dataset representation is $\mathbb{R}^{57}$, for the emoji models is $\mathbb{R}^{567}$, and, finally, for the keyword is $\mathbb{R}^{2048}$.

2.3. Configurations

We tested 13 different algorithms for each task. The configuration having the best performance was submitted to the contest. The best performance was computed using cross-validation, where the training set was split into a training (80%) and a validation set (20%).

The different configurations tested in this competition are described below. These configurations include BoW and a combination of BoW with dense representations. Stack generalization combines the different text classifiers, and the top classifier was a Naive Bayes algorithm. The specific implementation of this configuration can be seen in EvoMSA's documentation, particularly in the Section Competition. The implementation of the best configuration for each problem is also described in the appendix.

bow Pre-trained BoW where the tokens are selected based on a normalized frequency w.r.t. its type, i.e., bigrams, words, and q-grams of characters.

bow_voc_selection Pre-trained BoW where the tokens correspond to the most frequent ones.

bow_training_set BoW trained with the training set; the number of tokens corresponds to all the tokens in the set.

stack_bow_keywords_emojis Stack generalization approach where the base classifiers are the BoW, the emojis, and the keywords dense BoW.

stack_bow_keywords_emojis_voc_selection Stack generalization approach where the base classifiers are the BoW, the emojis, and the keywords dense BoW. The tokens in these models were selected based on a normalized frequency w.r.t. its type.

stack_bows Stack generalization approach where the base classifiers are BoW with the two token selection procedures described previously (i.e., bow and bow_voc_selection).

stack_2_bow_keywords Stack generalization approach with four base classifiers. These correspond to two BoW and two dense BoW (emojis and keywords), where the difference in each is the procedure used to select the tokens, i.e., the most frequent or normalized frequency.

stack_2_bow_tailored_keywords Stack generalization approach with four base classifiers. These correspond to two BoW and two dense BoW (emojis and keywords), where the difference in each is the procedure used to select the tokens, i.e., the most frequent or normalized frequency. The second difference is that the dense representation with normalized frequency also includes models for the most discriminant words selected by a BoW classifier in the training set. We refer to these latter representations as *tailored keywords*.

stack_2_bow_all_keywords Stack generalization approach with four base classifiers equivalent to stack_2_bow_keywords where the difference is that the dense representations include the models created with the human-annotated datasets.

stack_2_bow_tailored_all_keywords Stack generalization approach with four base classifiers equivalent to stack_2_bow_all_keywords, where the difference is that the dense representation with normalized frequency also includes the tailored keywords.

stack_3_bows Stack generalization approach with three base classifiers. All of them are BoW; the first two correspond pre-trained BoW with the two token selection procedures described previously (i.e., bow and bow_voc_selection), and the latest is a BoW trained on the training set (i.e., bow_training_set).

stack_3_bows_tailored_keywords Stack generalization approach with five base classifiers. The first corresponds to a BoW trained on the training set, and the rest are used in stack_2_bow_tailored_keywords.

stack_3_bow_tailored_all_keywords Stack generalization approach with five base classifiers. It is comparable to stack_3_bows_tailored_keywords being the difference in the use of the tailored keywords.

3. Results

Table 1 presents the performance, in terms of macro-F1, of the different configurations in a validation set. It also includes the performance of our submission in the competition and the competition's winner. It can be observed that the best configuration in the Type and Country problems is the BoW trained on the training set. This algorithm is described in [19] and implemented using B4MSA. On the other hand, the best configuration for Polarity corresponds to a stack generalization approach using as base classifiers two pre-trained BoW. It is interesting that for the whole competition, the best configurations found do not take advantage of any of the dense representations, which are the ones that incorporate additional information into the models. The extreme case is for Type and Country, where the best configuration is a BoW representation with a linear SVM.

Another comparison that one can make with the Table's data is computing the difference between the best performance in the validation set and the worst; it can be observed that for the Type, that difference is only 0.6%. For the Country, the difference is 3.4%, and for Polarity is 8.1%. In the competition's data, the Type is referred to the type of place (hotel, restaurant, and attractive), the Country could be Mexico, Cuba, and Colombia, and Polarity refers to the polarity level. For more details, please see the overview manuscript of the competition in [6].

The difference of 0.6% indicates that the performance of type is saturated, and following this approach will be complicated to improve. On the other problems, there might be room for improvement following the presented approach.

Table 1
The first block presents the performance, in terms of macro-F1, of the different configurations on the validation set. The second block shows the performance of our submission (INGEOTEC) and the competition's winner. The best performance is in boldface.

Configuration	Type	Country	Polarity
bow_training_set	**0.9802**	**0.9260**	0.5179
bow	0.9793	0.9194	0.5167
stack_3_bows	0.9793	0.9225	0.5603
bow_voc_selection	0.9792	0.9200	0.5152
stack_3_bow_tailored_all_keywords	0.9783	0.9166	0.5467
stack_3_bows_tailored_keywords	0.9783	0.9164	0.5448
stack_bows	0.9782	0.9167	**0.5605**
stack_2_bow_tailored_keywords	0.9773	0.9097	0.5448
stack_2_bow_tailored_all_keywords	0.9773	0.9101	0.5446
stack_2_bow_keywords	0.9769	0.9076	0.5420
stack_2_bow_all_keywords	0.9768	0.9076	0.5431
stack_bow_keywords_emojis	0.9743	0.8951	0.5310
stack_bow_keywords_emojis_voc_selection	0.9742	0.8949	0.5346
Competition			
Winner	**0.9903**	**0.9420**	**0.6217**
INGEOTEC	0.9805	0.9271	0.5549
Difference	1.0%	1.6%	12.0%

The second block in the table presents the performance of our participation as INGEOTEC in the competition and the performance of the competition's winner. The last row shows the difference in percentage between these two values. As can be seen, the difference in Type and Country is 1.0% and 1.6%, indicating that the BoW trained on the training set is exceptionally competitive with the winner. On the other hand, there is a difference of 12% in polarity; however, it is essential to note that the only additional information used was to estimate the IDF coefficients and to select the vocabulary (i.e., tokens).

4. Conclusions

Sentiment Analysis is a powerful tool that can be used by businesses and governments to understand people's opinions and emotions about various topics. In the tourism industry, Sentiment Analysis can be used to track customer satisfaction, identify trends, and improve marketing campaigns. We used our EvoMSA framework to merge different internal model outputs to solve the Rest-Mex 2023 challenge; these models are primarily large pre-trained and locally trained vocabularies capturing lexical and semantic features along with linear SVM. We obtained competitive models for all classification tasks compared with more complex and expensive deep learning approaches.

References

[1] M. Á. Álvarez-Carmona, R. Aranda, S. Arce-Cardenas, D. Fajardo-Delgado, R. Guerrero-Rodríguez, A. P. López-Monroy, J. Martínez-Miranda, H. Pérez-Espinosa, A. Y. Rodríguez-González, Overview of rest-mex at iberlef 2021: Recommendation system for text mexican tourism (2021).

[2] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, D. Fajardo-Delgado, R. Guerrero-Rodríguez, L. Bustio-Martínez, Overview of rest-mex at iberlef 2022: Recommendation system, sentiment analysis and covid semaphore prediction for mexican tourist texts, Procesamiento del Lenguaje Natural 69 (2022) 289–299.

[3] R. Guerrero-Rodriguez, M. Á. Álvarez-Carmona, R. Aranda, A. P. López-Monroy, Studying online travel reviews related to tourist attractions using nlp methods: the case of guanajuato, mexico, Current Issues in Tourism (2021) 1–16. doi:https://doi.org/10.1080/13683500.2021.2007227.

[4] M. A. Álvarez Carmona, R. Aranda, A. Y. Rodríguez-Gonzalez, D. Fajardo-Delgado, M. G. Sánchez, H. Pérez-Espinosa, J. Martínez-Miranda, R. Guerrero-Rodríguez, L. Bustio-Martínez, Ángel Díaz-Pacheco, Natural language processing applied to tourism research: A systematic review and future research directions, Journal of King Saud University - Computer and Information Sciences 34 (2022) 10125–10144. URL: https://www.sciencedirect.com/science/article/pii/S1319157822003615. doi:https://doi.org/10.1016/j.jksuci.2022.10.010.

[5] A. Diaz-Pacheco, M. A. Álvarez Carmona, R. Guerrero-Rodríguez, L. A. C. Chávez, A. Y. Rodríguez-González, J. P. Ramírez-Silva, R. Aranda, Artificial intelligence methods to support the research of destination image in tourism. a systematic review, Journal of Experimental & Theoretical Artificial Intelligence 0 (2022) 1–31. URL: https://doi.org/10.1080/0952813X.2022.2153276. doi:10.1080/0952813X.2022.2153276. arXiv:https://doi.org/10.1080/0952813X.2022.2153276.

[6] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, L. Bustio-Martínez, V. Muñiz-Sánchez, A. P. Pastor-López, F. Sánchez-Vega, Overview of rest-mex at iberlef 2023: Research on sentiment analysis task for mexican tourist texts, Procesamiento del Lenguaje Natural 71 (2023).

[7] M. Graff, S. Miranda-Jiménez, E. S. Tellez, D. Moctezuma, EvoMSA: A Multilingual

Evolutionary Approach for Sentiment Analysis, Computational Intelligence Magazine 15 (2020) 76 – 88. URL: http://arxiv.org/abs/1812.02307.

[8] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259. URL: https://www.sciencedirect.com/science/article/pii/S0893608005800231. doi:10.1016/S0893-6080(05)80023-1.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[12] M. Thelwall, Sentiment analysis for tourism, Big Data and Innovation in Tourism, Travel, and Hospitality: Managerial Approaches, Techniques, and Applications (2019) 87–104.

[13] C. Callejas-Hernández, E. Rivadeneira-Pérez, F. Sánchez-Vega, A. P. López-Monroy, E. Villatoro-Tello, The winning approach for the recommendation systems shared task @rest_mex 2022, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022, volume 3202 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3202/restmex-paper1.pdf.

[14] J. Vásquez, H. Gómez-Adorno, G. Bel-Enguix, Bert-based approach for sentiment analysis of spanish reviews from tripadvisor, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021, volume 2943 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 165–170. URL: https://ceur-ws.org/Vol-2943/restmex_paper6.pdf.

[15] M. Wang, Y. Yu, Deep sentiment analysis of the feelings expressed by tourists based on bert model, in: 2022 International Conference on Culture-Oriented Science and Technology (CoST), 2022, pp. 130–133. doi:10.1109/CoST57098.2022.00036.

[16] Z. Ma, C. Ye, H. Zhou, A study on sentiment analysis for smart tourism, in: 2022 International Conference on Service Science (ICSS), 2022, pp. 25–32. doi:10.1109/ICSS55994.2022.00014.

[17] B. Ray, A. Garain, R. Sarkar, An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews, Applied Soft Computing 98 (2021) 106935.

[18] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, Procesamiento

del Lenguaje Natural 71 (2023).
[19] E. S. Tellez, S. Miranda-Jiménez, M. Graff, D. Moctezuma, R. R. Suárez, O. S. Siordia, A simple approach to multilingual polarity classification in Twitter, Pattern Recognition Letters 94 (2017) 68–74. doi:10.1016/j.patrec.2017.05.024.

A. Library usage

This appendix aims to illustrate how the best configurations were implemented using EvoMSA [7]. The first step is to install the library, which can be done using the Anaconda package manager with the following instruction.

```
conda install -c conda-forge EvoMSA
```

A more general approach to installing EvoMSA is through the use of the command pip, as illustrated in the following instruction.

```
pip install EvoMSA
```

Once EvoMSA is installed, one must load a few libraries; the libraries used in the best configurations are the following.

```
from EvoMSA import BoW, DenseBoW
from EvoMSA import StackGeneralization as Stack
from EvoMSA.utils import Linear, b4msa_params
```

The configuration used in the type and country is a BoW trained on the training set. The first step is to load the default parameters (line 2), then on line 3 and 4, removes the parameters that limit the vocabulary size. In line 5, the text classifier is built, and the predictions are computed in the last instruction. The arguments *ts* and *vs* correspond to the training and validation (gold) sets. These variables are a list of dictionaries, where the text is in the key *text* and the label is the key *klass*.

```
def bow_training_set(lang, tr, vs):
    params = b4msa_params(lang=lang)
    del params['token_max_filter']
    del params['max_dimension']
    bow_no_pre = BoW(lang=lang,
                     pretrain=False,
                     b4msa_kwargs=params).fit(tr)
    return bow_no_pre.predict(vs)
```

The configuration used in polarity corresponds to a stack generalization approach where the base classifiers are the BoW models. The first BoW model is initialized in line 2, the second BoW can be seen in line 3, and the stack-generalization method is in line 4. The last instruction corresponds to the predictions in the validation set.

```
def stack_bows(lang, tr, vs):
    bow = BoW(lang=lang)
    bow2 = BoW(lang=lang, voc_selection='most_common')
```

```
    stack = Stack(decision_function_models=[bow, bow2]).fit(tr)
    return stack.predict(vs)
```