String Kernels for Polarity Classification: A Study Across Different Languages

Rosa M. Giménez-Pérez[1(✉)], Marc Franco-Salvador[2], and Paolo Rosso[1]

[1] Universitat Politècnica de València, Valencia, Spain
{rogipe2,prosso}@upv.es
[2] Symanto Research, Nuremberg, Germany
marc.franco@symanto.net

Abstract. The polarity classification task has as objective to automatically deciding whether a subjective text is positive or negative. Using a cross-domain setting implies the use of different domains for the training and testing. Recently, string kernels, a method which does not employ domain adaptation techniques has been proposed. In this work, we analyse the performance of this method across four different languages: English, German, French and Japanese. Experimental results show the strong potential of this approach independently from the language.

Keywords: String kernels · Sentiment analysis · Single-domain Cross-domain

1 Introduction

Since the Web 2.0 emergence, the Internet has been providing different channels where people can express their opinions about many products and services. As a result, blogs, fora and social media have become an important information source for companies. As a consequence, there is a high interest in identifying opinions and reviews in order to improve the business they offer.

The task of deciding if those texts are positive or negative depending on the overall sentiment detected is known as sentiment or polarity classification task. Single-domain polarity classification (SD) [6] refers to the standard text classification setting. Cross-domain polarity classification (CD) [2] refers to testing on a different domain from that or those used for training the model. Although this task can be tackled as a common text classification problem, sentiment may be expressed in a more subtle manner. Furthermore, in the CD variant it exists the additional difficulty of using different vocabularies among the domains. For these reasons, solutions based only on bag-of-words representations are not enough.

In [4] the authors studied the performance of string kernels at SD and CD level for English texts with very promising results and showed their capability to capture the lexical peculiarities that characterise the polarity in a domain-independent way.

M. Silberztein et al. (Eds.): NLDB 2018, LNCS 10859, pp. 489–493, 2018.
https://doi.org/10.1007/978-3-319-91947-8_50

In order to further investigate string kernels performance in polarity classification, in this work we study their application for other languages, i.e., English, German, French and Japanese. This study includes the analysis of the impact of key parameters such as the string length depending on the alphabet employed. This is, to the best of our knowledge, the first time that this dataset has been used in the mono-lingual polarity classification task.

2 String Kernels

String Kernels (SK) are functions that measure the similarity of string pairs at lexical level. Their dual representation allows to keep the feature space reduced, even working with a huge number of characteristics. Following the implementation and formulation of Ionescu et al. [5],[1] the p-grams kernel function counts how many substrings of length p have two strings s and t in common: $\mathrm{k}_p(s,t) = \sum_{v \in L^p} f(\mathrm{num}_v(s), \mathrm{num}_v(t))$, where $\mathrm{num}_v(s)$ is the number of occurrences of string v as a substring of s, p is the length of v, and L is the alphabet used to generate v. The function $f(x,y)$ varies depending on the type of kernel: $f(x,y) = x \cdot y$ in the p-spectrum kernel; $f(x,y) = \mathrm{sgn}(x) \cdot \mathrm{sgn}(y)$ in the p-grams presence bits kernel; $f(x,y) = \min(x,y)$ in the p-grams intersection bits kernel.

Taking into account that the spectrum kernel provides the highest values and presence kernel the lowest, this can gives us an idea about what these kernels capture. The spectrum kernel offers high values even when the texts are only partially related. The intersection kernel employs the n-gram frequency to provide with a precise lexical similarity measure. Finally, the presence kernel captures the lexical *core meaning* of the texts by smoothing the n-gram repetitions.

The kernels we implemented combine different n-gram lengths by adding the kernel values obtained for each n (see Sect. 3.2 for details about our parameter selection) and are normalised as follows: $\hat{\mathrm{k}}(s,t) = \mathrm{k}(s,t)/\sqrt{\mathrm{k}(s,s) \cdot \mathrm{k}(t,t)}$.

We perform the classification with Kernel Discriminant Analysis (KDA) [1], We compute the feature matrices $Y = KU$ and $Y_\mathrm{t} = K_\mathrm{t}U$, where K and K_t are the training and test instance kernels. For each class c, we create the prototype Y_c as the average of all vectors of Y that correspond to the instances of class c. Finally, we classify each test instance by identifying the class of the prototype with the lowest mean squared error between $Y_\mathrm{t}(i)$ and Y_c.

3 Evaluation

3.1 Dataset and Tasks Setting

We employ the CLS dataset[2] which is formed by Amazon product reviews of three domains (Books, DVDs and Music) in four languages: English, German,

[1] http://string-kernels.herokuapp.com/.

[2] https://www.uni-weimar.de/de/medien/professuren/medieninformatik/webis/data/webis-cls-10/.

French and Japanese. Each language-domain partition is formed by a training and a test set with 1,000 positive and 1,000 negative reviews each one.

To evaluate our approach, we use the presence ($k_p^{0/1}$), intersection ($k_p^{\cap}$), and spectrum (k_p) kernels. In order to compare the results, we calculate a baseline using the Tf-idf weighting and Support Vector Machines (SVM) using a linear kernel. In addition, we implement a model based on distributed representations of words. We use the recent Facebook's pretrained FastText [3] vectors.[3] We average the word vectors to represent the instances. We classify using SVM with a linear kernel. In this work, the best results of the tables are highlighted in bold.

3.2 String Kernel Parameter Selection

The kernel n-gram length and the KDA's regularisation factor α are adjusted with a 10-cross-validation over the train partition of each domain. First, we set α to a default value (0.2) and select the presence kernel to analyse the results for different combinations of n-gram lengths. Following the procedure of Giménez-Pérez et al. [4] we tried combinations for $4 \leq n \leq 9$ for all the evaluated languages. However, during the prototyping, we realised that n-grams within that range are not adequate for languages such as Japanese. The lexical and semantic information included inside a symbol of its alphabet is notably larger than the information included in the Roman alphabet. Therefore, we modified the search space of the Japanese string length to $1 \leq n \leq 6$. Once that parameter was adjusted, we tested different α values between 0.01 and 1.0 and selected the best for every language, domain and kernel in each task.

3.3 Results

First, we evaluate and compare the models at SD level. SK outperform the other two models in all the cases. This manifests their potential for texts written using the Roman and Japanese alphabets. The French Books domain and the English DVDs one work marginally better with the spectrum kernel. The English Books domain shows the best results with the intersection kernel. Excepting those cases, all the other domains and languages obtained the best results using the presence kernel. Giménez-Pérez et al. [4] proved that this method offers a notable stability among different English domains. That statement is extended here to the rest of languages evaluated.

For CD level, we train with all the domains but the one used in the test partition. Similarly to the single-domain results, SK outperform the two compared models. This reinforces the SK suitability regarding their potential with the Roman and Japanese alphabets. Although the best results are obtained on average with the presence and intersection kernel, the spectrum kernel also obtains competitive results, being even the best option in some cases (French Books and Japanese DVDs domains). Results are shown in Table 1.

[3] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md.

Table 1. SD and CD polarity classification accuracy (in %).

	Language	EN			GE			FR			JP		
	Method	Books	DVDs	Music	Books	DVDs	Music	Books	DVDs	Music	Books	DVDs	Music
SD	Tf-idf+SVM	81.4	80.7	81.6	83.2	81.2	81.3	84.1	84.0	85.6	77.4	79.7	79.2
	FT+SVM	79.6	77.8	78.8	79.6	79.3	79.2	80.3	79.9	77.8	73.4	73.8	75.5
	SK($k_p^{0/1}$)	82.6	82.4	**82.9**	**86.4**	**83.2**	**84.5**	84.6	**85.3**	**86.3**	**80.4**	**81.9**	**81.6**
	SK($k_p^{\cap}$)	**82.8**	83.0	82.7	86.3	82.8	84.1	84.3	85.0	86.0	80.1	81.8	80.8
	SK(k_p)	82.4	**83.2**	81.8	85.5	81.9	84.0	**84.8**	84.8	86.0	80.2	80.1	80.7
CD	Tf-idf+SVM	78.7	79.3	80.2	80.7	80.6	80.2	**82.4**	83.3	81.4	77.3	77.4	78.7
	FT+SVM	80.0	75.7	77.8	79.0	75.4	77.9	78.2	79.5	77.0	71.3	73.4	73.7
	SK($k_p^{0/1}$)	81.4	81.5	**81.8**	**84.4**	81.4	**83.6**	82.2	84.4	85.4	**79.9**	80.6	**81.1**
	SK($k_p^{\cap}$)	**81.5**	**81.6**	81.5	84.0	**82.5**	82.7	82.1	**84.5**	**85.5**	79.8	80.5	80.9
	SK(k_p)	79.9	81.0	81.7	82.6	81.3	82.4	**82.4**	83.6	84.7	79.4	**80.9**	79.2

4 Conclusions

In this paper we studied the use of string kernels for the single and cross-domain polarity classification task and studied their behaviour across four languages: English, German, French and Japanese. We used for the first time the CLS dataset in mono-lingual polarity classification tasks. We evaluated the intersection, presence and spectrum kernels when classifying with kernel discriminant analysis. We evaluated the importance of the n-gram length selection depending on the language. This showed that the best results for the Japanese alphabet are obtained when selecting smaller lengths than the ones employed with the Roman alphabet. The best classification results were obtained on average using the presence and intersection kernel. In addition, the stability of the results among the different evaluated domains was notably high for all the evaluated languages. Finally, string kernels showed strong potential, in all the evaluated languages, at capturing the lexical peculiarities that characterise polarity in a domain-independent way.

Acknowledgements. The work of the third author was partially funded by the Spanish MINECO under the research project SomEMBED (TIN2015-71147-C2-1-P).

References

1. Baudat, G., Anouar, F.: Generalized discriminant analysis using a Kernel approach. Neural Comput. **12**(10), 2385–2404 (2000)
2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Proceedings of the Annual Meeting of the Association of Computational Linguistics, pp. 440–447 (2007)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5** (2017)
4. Giménez-Pérez, R.M., Franco-Salvador, M., Rosso, P.: Single and cross-domain polarity classification using string kernels. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, p. 558 (2017)

5. Ionescu, R.T., Popescu, M., Cahill, A.: Can characters reveal your native language? A language-independent approach to native language identification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1363–1373 (2014)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)