INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

TOTAL FREEDOM IN MANUAL TEXT CLASSIFICATION SUPPORTED BY UNOBTRUSIVE MACHINE LEARNING

[1]B. Revathi, Pusa pelly Akhila, [2]Reddyvari Bhargavi Reddy, [3]Sarabu Jyothika, [4]Yalam Sarvani Reddy

[1]Assistant Professor, Department of CSE(DS), Malla Reddy Engineering College for Women (Autonomous Institution – UGC, Govt. of India), Hyderabad, INDIA.

[2345]UG,Department of CSE(DS), Malla Reddy Engineering College for Women (Autonomous Institution – UGC, Govt. of India), Hyderabad , INDIA.

ABSTRACT

We present the Interactive Classification System (ICS), a web-based application that supports the activity of manual text classification. The application uses machine learning to continuously fit automatic classification models that are in turn used to actively support its users with classification suggestions. The key requirement we have established for the development of ICS is to give its users total freedom of action: they can at any time modify any classification schema and any label assignment, possibly reusing any relevant information from previous activities. We investigate how this requirement challenges the typical scenarios faced in machine learning research, which instead give no active role to humans or place them into very constrained roles, e.g., on-demand labeling in active learning processes, and always assume some degree of batch processing of data. We satisfy the ‘‘total freedom’’ requirement by designing an unobtrusive machine learning model, i.e., the machine learning component of ICS acts as an unobtrusive observer of the users, that never interrupts them, continuously adapts and updates its models in response to their actions, and it is always available to perform automatic classifications. Our efficient implementation of the unobtrusive machine learning model combines various machine learning methods and technologies, such as hash-based feature mapping, random indexing, online learning, active learning, and asynchronous processing.

I. INTRODUCTION The task of text classification consists of selecting labels that are relevant to the content of a document. This label assignment process

ISSN: 2057-5688

gives an explicit and structured form to the information that is latent and represented in an unstructured way in the text. Classification enables the successive use of information processing/mining tools that otherwise would not be directly applicable to the original information represented using natural language. For example, it is possible to classify a stream of social posts to mark those relevant to a certain political topic, doing it for a set of topics emerging from an ongoing political debate. The availability of this classification enables to perform various tasks on the data, e.g.: to measure the variation of the engagement of the public over time and topics in order to identify which are the most relevant ones; to profile users' interests, possibly targeting each different profile with different messages; to select the content that is relevant to one specific topic in order to perform further analysis, e.g., sentiment analysis. The classification of documents is an intellectual task that requires giving a semantic to the concepts represented by the labels and recognizing such concepts in the content of documents. Some concepts may be simple to define and recognize, e.g., the mention of a brand name, others may be much harder to give a clear and shared definition, e.g., the expression of sarcasm. With the exception of trivial tasks, i.e., those that can be solved by simple string matching, the effort required to read, understand the document, and match it to the relevant labels makes classification performed by humans a low-productivity activity that is expensive to scale. This is why the automatic classification of texts is a research topic that has a long history in computer science [1].

The rest of the paper is organized as follows. Section II presents the related work, framing the context of our work and comparing ICS with similar existing systems. Section III describes the interfaces and functionalities provided by ICS to the users. Section IV details on the architecture and implementation of ICS, with a special attention on the machine learning component of ICS. In Section V we present experiments that evaluate the machine learning component of ICS, on four relevant classification problems (i.e., singlelabel classification, multi-label classification, binary sentiment classification, transfer

ISSN: 2057-5688

learning). Conclusions are drawn in Section VI.

II. RELATED WORK

This paper follows the terminology currently used in machine learning research. Yet, given that its subject can be of interest to a diverse audience we will briefly discuss here the terms used over time and across disciplines to refer to the process of labeling documents according to a classification schema, before moving on to the specific topics of our work. The activity of labeling documents has been called in many different ways: text filtering [6]–[8], text routing [9], text categorization [10]–[12], text classification [13]–[15], text coding [16], [17]. The terms filtering and routing somewhat imply the goal of the classification process, i.e., to filter out non-relevant documents or to route documents to different processing channels depending on their content. However, most papers that use these terms just focus on the accurate assignment of labels to documents, with no actual interest in the subsequent processing. The other three terms do not denote any assumption about the use of the assigned labels. In the domain of computer science, and especially in machine learning, the terms categorization and classification are used almost as synonyms. The term classifier is typically used to denote an actual instance of an automatic method that assigns labels, while the terms category and class are the ones most used to refer to the concept and properties represented by a label. The term label is often used as a synonym for the term category, especially when defining the constraints on how to assign labels, i.e.: in a single-label classification, a document must be assigned with one and only one label from the set of available labels, in a multi-label classification a document can be assigned with zero, one, or more labels. The last term, coding, is more frequently used in social sciences and market research, fields in which the classification activity is mostly carried out by humans (called coders) and rarely by means of automatic method

III METHODOLOGY

DATASETS

A dataset is a dynamic set of documents. The collection of documents forming a dataset can change over time, e.g.,

ISSN: 2057-5688

adding new tweets from a running filtered stream, even when classification is already ongoing. A user can perform manual classification of a dataset in two ways: ‘‘browse and code’’, and ‘‘live classification’’.

Documents can be added to a dataset in batches or single instances. The web application interface allows uploading a CSV file, in which a document is represented as an external unique identifier (to link it to its external source) and its text. The web service API has methods to upload a CSV file or to send text data directly in the POST request. This second method is used for example by a Twitter filtered stream script included in the package, which continuously collects tweets from Twitter and populates datasets on ICS.

LABEL SUGGESTION

As mentioned above, the machine learning component of the system can provide users with suggestions of which labels should be assigned to a document for the currently selected classifiers. These suggestions come in the form of a symbol and a color hint on the suggested labels (see Figures 3 and 4). Suggestions are always available, independently of which browsing mode is in use. Whenever an automatic classifier is updated, any suggestion produced by that automatic classifier that is currently shown to any user is updated accordingly to the output of the updated version. The user is not required to act any differently than when label suggestions are not shown. When label suggestions are shown, the interface also shows a bar with the history of agreements/disagreements and the agreement percentage (i.e., accuracy) over the last n labeling actions (see Figure 2, lower part).

IV IMPLEMENTATION

The scenario of use we set for ICS imposes relevant constraints on both the text indexing and the machine learning components. The machine learning algorithm, which is described in the next Section, works with the vector representations of text used by most statistical machine learning methods, i.e., real-valued, high dimensional vectors $x \in V = R^n$ where V is a high dimensional vector space. The transformation of a document into a vector consists of two phases: feature

ISSN: 2057-5688

extraction and indexing. The feature extraction process identifies in the text all the linguistic features (e.g., words, lemmas, n-grams, PoS, entities. . .) that may result in useful information for the machine learning algorithm. The set D of all the linguistic features observed in the training set by feature extraction is called the feature space.

A basic implementation of LRI uses a dictionary that maps each feature to its random vector. Whenever a feature is extracted from text, the dictionary is checked to retrieve the random vector, if it is missing a random vector is generated by means of some random number generator and added to the dictionary for future use. This implementation has a memory cost, to store the dictionary, i.e., feature-vector pairs, and a computational cost, to retrieve the vector given a feature. The more efficient implementation replaces the use of an explicit, memorized dictionary, with the use of an implicit dictionary based on hashing functions, a method known as feature hashing [8], or hashing trick. Feature hashing determines the vector representing a feature by means of a hashing function that takes in input the representation of the feature and returns a numeric value. That numeric value is then mapped, typically via modulus operation, to a dimension in the vector space, in which a +1 or −1 value (the sign is also determined from the hash) is set. Feature hashing removes the need to store the dictionary and is in theory able to handle feature spaces of infinite size. Algorithm 1 shows the pseudo-code that implements the LRI method based on feature hashing. Note that the hashing function is called twice, to determine the two non-zero dimensions of the random vector, using two different seeds, as different seeds determine completely independent outputs.

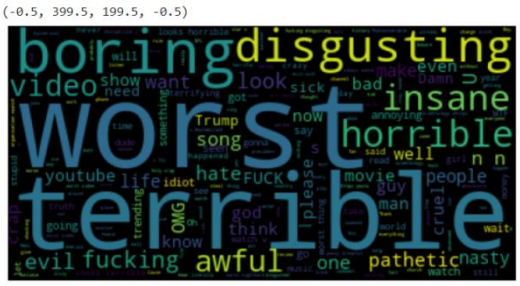

Fig 1: Text words

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

Fig 2: Emoji classification

V CONCLUSION

We presented ICS, a web-based application that supports the activity of manual text classification. ICS has been designed and implemented to give its users total freedom of action. This is an innovation with respect to the typical approach of machine learning research applied to text, which focuses on the algorithms, and assigns the human actors to constrained roles within the workflow of the algorithm. Online learning methods, especially when coupled with active learning, do give some freedom to their users. They also bring in the advantage of having usable models for prediction since the early steps of training set construction. Yet, they still do not cover the additional freedom of action we require for our system, e.g., adding/removing labels, and merging existing models to define new, different models that leverage the already acquired supervised information. To implement such solutions, we combined the flexibility of online learning methods with additional theoretical and technological tools, such as feature hashing, random indexing, and asynchronous processing. The resulting system satisfies our requirements, and also shows new avenues for the development of classification systems.

VI REFERENCES

[1] M. E. Maron, ‘‘Automatic indexing: An experimental inquiry,’’ J. ACM,vol. 8, no. 3, pp. 404–417, Jul. 1961, doi: 10.1145/321075.321084.

[2] C. C. Aggarwal and C. Zhai, A Survey of Text Classification Algorithms. Heidelberg, Germany: Springer, 2012, pp. 163–222, doi: 10.1007/978-1-4614-3223-4_6.

[3] F. Sebastiani, ‘‘Machine learning in automated text categorization,’’ACM

ISSN: 2057-5688

Comput. Surv., vol. 34, no. 1, pp. 1–47, 2002, doi:10.1145/505282.505283.

[4] B. Settles, ‘‘Active learning literature survey,’’ Dept. Comput. Sci., Univ.Wisconsin-Madison, Wisconsin, WI, USA, Tech. Rep. TR1648, 2009.[Online]. Available: http://digital.library.wisc.edu/1793/60660

[5] A. Esuli, ‘‘Interactive classification system,’’ Zenodo, 2022, doi:10.5281/zenodo.6586244.

[6] D. W. Oard, ‘‘The state of the art in text filtering,’’ User Model. UserAdapted Interact., vol. 7, no. 3, pp. 141–178, Jun. 1997.

[7] R. E. Schapire, Y. Singer, and A. Singhal, ‘‘Boosting and Rocchio applied to text filtering,’’ in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop.Inf. Retr., vol. 98, Aug. 1998, pp. 215–223, doi: 10.1145/290941.290996.

[8] I. Soboroff and C. K. Nicholas, ‘‘Combining content and collaboration in text filtering,’’ in Proc. Workshop Mach. Learn.Inf. Filtering (IJCAI), 1999, pp. 86–91. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.8019

[9] S. Wermter, ‘‘Neural network agents for learning semantic text classification,’’ Inf. Retr., vol. 3, no. 2, pp. 87–103, 2000, doi: 10.1023/A:1009942513170.

[10] T. Joachims, ‘‘Text categorization with support vector machines: Learning with many relevant features,’’ in Proc. Eur. Conf. Mach. Learn., 1998, pp. 137–142, doi: 10.1007/BFb0026683.

[11] Y. Yang and X. Liu, ‘‘A re-examination of text categorization methods,’’ in Proc. 22nd Annu. Int. ACM Conf. Res. Develop. Inf. Retr. (SIGIR), 1999, pp. 42–49, doi: 10.1145/312624.312647.

[12] R. Johnson and T. Zhang, ‘‘Semi-supervised convolutional neural networks for text categorization via region embedding,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015, pp. 919–927. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/acc3e0404646c57502b480dc052c4fe1-Abstract.html

[13] A. McCallum and K. Nigam, ‘‘A comparison of event models for naive Bayes text classification,’’ in Proc. AAAI Workshop Learn. Text

ISSN: 2057-5688

Categorization, vol. 752, no. 1, 1998, pp. 41–48.

[14] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, ‘‘Text classification from labeled and unlabeled documents using EM,’’ Mach. Learn., vol. 39, nos. 2–3, pp. 103–134, 2000, doi: 10.1023/A:1007692713085.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, ‘‘Bag of tricks for efficient text classification,’’ in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, vol. 2, 2017, pp. 427–431. [Online]. Available:ttps://aclanthology.org/E17-2068

[16] D. J. Hruschka, D. Schwartz, D. C. S. John, E. Picone-Decaro, R. A. Jenkins, and J. W. Carey, ‘‘Reliability in coding open-ended data: Lessons learned from HIV behavioral research,’’ Field Methods, vol. 16,no. 3, pp. 307–331, Aug. 2004, doi: 10.1177/1525822X04266540.

[17] M. Cope, ‘‘Coding transcripts and diaries,’’ Key Methods Geography,vol. 440, pp. 440–452, Jan. 2010.

[18] Y. Fu, X. Zhu, and B. Li, ‘‘A survey on instance selection for active learning,’’ Knowl. Inf. Syst., vol. 35, no. 2, pp. 249–283, May 2013, doi: 10.1007/s10115-012-0507-8.