Cloud-based textual analysis as a basis for document classification

George R S Weir[1], Kolade Owoeye, Alice Oberacker and Haya Alshahrani
Department of Computer & Information Sciences
University of Strathclyde
Glasgow, UK
{george.weir;kolade.owoeye}@strath.ac.uk, oberackera@googlemail.com, hayaalshahrani@yahoo.com

Abstract— **Growing trends in data mining and developments in machine learning, have encouraged interest in analytical techniques that can contribute insights on data characteristics. The present paper describes an approach to textual analysis that generates extensive quantitative data on target documents, with output including frequency data on tokens, types, parts-of-speech and word n-grams. These analytical results enrich the available source data and have proven useful in several contexts as a basis for automating manual classification tasks. In the following, we introduce the Posit textual analysis toolset and detail its use in data enrichment as input to supervised learning tasks, including automating the identification of extremist Web content. Next, we describe the extension of this approach to Arabic language. Thereafter, we recount the move of these analytical facilities from local operation to a Cloud-based service. This transition, affords easy remote access for other researchers seeking to explore the application of such data enrichment to their own text-based data sets.**

Keywords-data mining; textual analysis; classification; feature-set; Cloud-service; Posit.

I. Introduction

As diverse sources of data are increasingly being gathered to create large pools of potential resource, techniques for analysis that facilitate new insights and added value to the raw data are sought with enthusiasm. In this setting, our previous work on text analysis has proven useful as a basis for enrichment of textual data.

The present paper outlines the context of text mining and classification before describing quantitative text analysis using the Posit textual analysis toolset. Thereafter, we detail the application of Posit to the classification of text data. This is followed by an account of Posit as a Cloud-based facility and the potential benefits this affords for distributed third-party application. We conclude with a description of on-going developments to extend the available features in the Cloud-Posit system.

II. Text mining and classification

Commonly, two main classes of text categorisation are recognised [1, 2]: text clustering and text classification. The first deals with finding a structure of groups within a given dataset, while the latter is given a set of groups against which each text is to be assigned. Moreover, the task of text classification is subjective in a way that human and machine might disagree on the classification of the data. Text classification can be single-labelled meaning every document is as- signed a single category, or multi-labelled in which case a document can be assigned to several possible categories. This method has the advantage of giving the user the possibility of a final decision to their own subjective opinion as several texts can be closely related to multiple categories. Several applications such as spam filtering, webpage classification, author- ship attribution or genre classification can be decided with text classifications. Among the various machine learning algorithms that have been used to build classifiers, [2] claims the ones that proved most successful in recent years are support vector machines (SVM) and boosting. SVM is a type of classification model, boosting, however, combines the decisions of a group of classifiers in order to achieve a better overall classification [2]. [3] agrees on the effectiveness of SVM, but also points out that this approach might find a suboptimal decision threshold for categories with low occurrences.

However, it remains a challenge to achieve high accuracy for all possible contexts at once, as no algorithm is most effective on all applications [4, 5]. Moreover, the labelling of the documents defines a bottleneck for every supervised classification method as it has to be done manually.

To solve this problem [6] developed a system to hierarchically classify unlabelled data. As already mentioned, classifying data manually is extremely expensive and slows the classification process down. Additionally, it grows to be an inefficient approach as with larger datasets the number of categories can exceed to thousands, of which each needs to be represented by a sufficient number of labelled documents. The

[1] Corresponding author

system solves this issue by using ontological knowledge and by searching 'pseudo-relevant documents on the Web' [6]. With the ontology it is possible to create a hierarchical model including the context of ancestors among different classes.

[3] compared the accuracy of SVM, k-Nearest Neighbours (k-NN) and Rocchio-Style Prototype Classifier with each other on the Reuters Corpus Vol. 1. Two variants of SVM were used. The first one was trained for each category by using the default settings and the latter tried to find optimal settings to improve results for unbalanced classes for each category and was trained using a leave-one-out cross validation. Results show that the first SVM classifications achieve the best F1 values, followed closely by the second SVM approach. k-NN and Rocchio-Style did not achieve as good results, which underlines the statement made by [2].

Another study by [4] compared results of k-NN, Rocchio-Style and Linear Least Square Fit (LLSF) with each other. Throughout the experiment k-NN achieved the best classification results, with Rocchio and LLSF showing reasonable efficiency. [4] however, states that SVM methods can be used to improve upon the k-NN results. The k-Nearest Neighbour method is a lazy learning method, because few calculations are done during the training phase. During the classification the distances to all training samples have to be calculated to find the k nearest samples, which makes it a lazy learning method and therefore more sensitive to noisy data as it only considers a few samples to make a decision [7, chap. 4].

For classifying text corpora, one has to develop an internal representation for the learning algorithms. The most common approach represents each text as a vector in which every position displays the existence of a word (set of words). Similar techniques do not only acknowledge the existence but also the frequency of words (bag of words) [8, 1]. The representation usually has a large number of features due to the number of unique words in the document. Therefore, it is appropriate to remove irrelevant features to optimise the prediction [5]. However, it needs to be shown if feature selection plays an important role when using the Posit toolset, as the number of features that can be extracted from the computed quantitative data do not expand the runtime of the learning algorithms drastically. As it is suggested in multiple papers [5, 8, 9] feature selection can improve the performance of classifiers.

[8] for example, points out that words with low frequencies can be neglected as well as so called stop words, such as 'a' and 'or'. However, for every approach one needs to bear in mind the possibly varying size of documents as the occurrences need to be normalised over the size of text. [5] suggest that the most suitable classification performance metrics is the receiver operating characteristic (ROC), which plots sensitivity against 1−specificity. The area under curve (AUC) can then be used to differentiate between perfect classification (AUC=1), classification by chance (AUC=0.5) and inverse classification (AUC=0). The advantage of this metric is its insensitivity to unbalanced categories.

In the field of computational linguistics n-grams are defined as a sequence of characters or words of length n. The Posit tool makes it possible to extract word grams of length 2, 3 and 4 including their frequency. Statistical features about word n-grams have been appropriated for text classification by [10]. The n-gram language model is handled in a similar way to a Naïve Bayes model. Each category is trained with a language model and every document can be evaluated on each of those models to decide to which it agrees the most. In this experimental paper it was shown that statistical data of n-grams can be used for a chain augmented Naïve Bayes classifier. An optimal size for n-grams can be found to improve the classification of documents. This underlines the likely importance of n-grams data, as it seems to be a source for improving upon classification accuracy.

III. Quantitative Text Analysis

The most popular approach to text classification represents each text as a vector of word occurrences (set or bag of words) [4, 8, 11, 1]. One way of modelling such a vector is denoting the occurrence of a word by setting the position to 1 and otherwise to 0. There are other models which also include the frequency of which a word appears in the text which can be of great importance to the classification. However, these approaches require a lot of computational time and optimisation, for example, using feature selection.

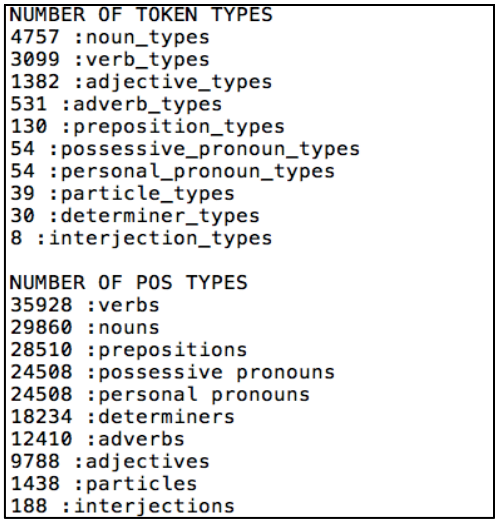
Another approach, which enriches the representation of texts for machine learning models is described in this section. Instead of representing a text with its words, we may calculate quantitative and statistical values for a text and use these features as a basis for classification.

The Posit Text Profiling Toolset [12, 13] offers a thorough quantitative analysis of an arbitrarily large text corpus with highly customisable features. Posit applies a Part-of-Speech tagger and outputs statistical details of the text content in terms of individual words (tokens) and word types. This frequency data is also provided for specific parts of speech, including frequency ordered details of each specific word in an analysed text. Significantly, in affording a basis for comparison between samples of text data, Posit's summary details can be employed as a feature set for use in classification of textual data.

The summary data output from Posit includes values for total words (tokens), total unique words (types), type/token ratio, number of sentences, average sentence length, number of characters, average word length, noun types, verb types, adjective types, adverb types, preposition types, personal pronoun types, determiner types, possessive pronoun types, interjection types, particle types, nouns, verbs, prepositions, personal pronouns, determiners, adverbs, adjectives, possessive pronouns, interjections, and particles. This comprises 27 features in all. An example of such output is shown in Figure 1.

When analysing texts using Posit, output is generated at several levels of detail. Of these, the summary level is the most general, e.g., the total number of verbs, nouns, adjectives, etc. (Figure 1). Two more detailed levels of output are provided: an intermediate (aggregate) part-of-speech analysis, and a finely detailed word types against parts-of-speech account.

At the intermediate level, frequency data is provided for the contents of the analysed text in terms of specific parts-of-speech, for example, types of verb: the base form of verbs, the gerund form, the past tense form, the past participle form, the 3rd person present form, the present tense (non-3rd person) form and the modal auxiliary form. An illustration of this intermediate level is shown in Figure 2.

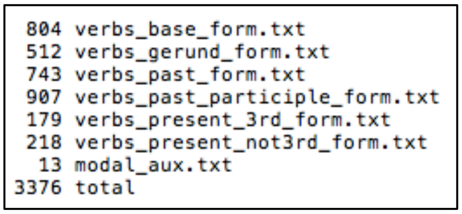


```
NUMBER OF TOKEN TYPES
4757 :noun_types
3099 :verb_types
1382 :adjective_types
531 :adverb_types
130 :preposition_types
54 :possessive_pronoun_types
54 :personal_pronoun_types
39 :particle_types
30 :determiner_types
8 :interjection_types

NUMBER OF POS TYPES
35928 :verbs
29860 :nouns
28510 :prepositions
24508 :possessive pronouns
24508 :personal pronouns
18234 :determiners
12410 :adverbs
9788 :adjectives
1438 :particles
188 :interjections
```

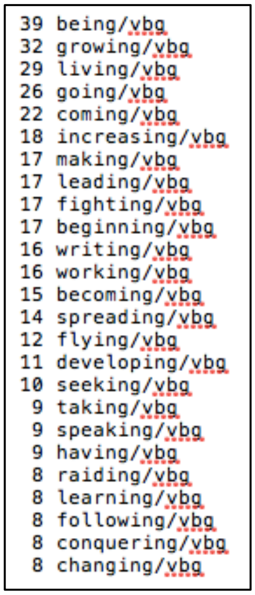
Figure 1: Example Posit summary output

At the fine detail level, frequency data is provided for each word in terms of part-of-speech type, for example, the number of occurrences of every word that is a verb of gerund form. An illustration of this fine detail level is shown in Figure 3.



```
 804 verbs_base_form.txt
 512 verbs_gerund_form.txt
 743 verbs_past_form.txt
 907 verbs_past_participle_form.txt
 179 verbs_present_3rd_form.txt
 218 verbs_present_not3rd_form.txt
  13 modal_aux.txt
3376 total
```

Figure 2: Example Posit aggregate output



```
39 being/vbg
32 growing/vbg
29 living/vbg
26 going/vbg
22 coming/vbg
18 increasing/vbg
17 making/vbg
17 leading/vbg
17 fighting/vbg
17 beginning/vbg
16 writing/vbg
16 working/vbg
15 becoming/vbg
14 spreading/vbg
12 flying/vbg
11 developing/vbg
10 seeking/vbg
 9 taking/vbg
 9 speaking/vbg
 9 having/vbg
 8 raiding/vbg
 8 learning/vbg
 8 following/vbg
 8 conquering/vbg
 8 changing/vbg
```

Figure 3: Example Posit fine detail output

IV. CLASSIFICATION USING POSIT

Since the basis of any classification is the ‘matching’ of features present across data samples, the feature set produced when texts are analysed using Posit provides a ready characterization of texts that can be contrasted for the purpose of classification. In our classification work to date, we have used only the summary output produced by Posit as the basis for a feature set that characterises each data sample.

To this end, [14] applied the Posit tool to generate summary output for data retrieved by the Terrorism and Extremism Network Extractor (TENE) web crawler [15]. This data had been manually classified into the categories ’pro-extremist’, ’neutral’ and ’anti-extremist’. Posit was applied in order to provide the quantitative syntactic features that ‘enrich’ the information given by the text corpus.

When used for classification with the J48 algorithm, the Posit approach matched 91.4% of the manually classified webpages correctly. An improved result of 95.3% correctly classified texts was accomplished with a Random Forest algorithm. These results led us to believe that through application of Posit analysis we could provide enriched insight on the content of textual data and afford effective classification of such data. The advantage of a quantitative approach, opposed to a vector representation of the existence of words in the text (bag of words), is that the number of features is much lower. Instead of dealing with millions of features [16], the Posit tool extracts 27 distinct values. Further research is underway to explore the extension of the Posit feature set, including frequency data on word combinations (n-grams) and frequency ratios (e.g., ratio of common nouns to proper nouns).

Following this effective application of Posit to the classification of extremist Web content, a similar approach was adopted with a dataset containing drug related texts from the Dark Web. Some of this data were manually classified as drugs-related positive or negative. A total of 1,245,410 texts were included in the initial set and this was reduced to 798,684 textual data items after cleaning. In the final data set, 91,088 items were pre-classified as drugs-related or not drugs-related.

A series of experiments using Posit-based classification were performed on this Dark Web data set, aiming to match against the training set provided by the manually classified subset of data. The results (Table 1) show that the K Nearest Neighbour algorithm (where k=1) gave the best performance (with an F-measure of 0.995), closely followed by the J48 algorithm (with an F-measure of 0.99).

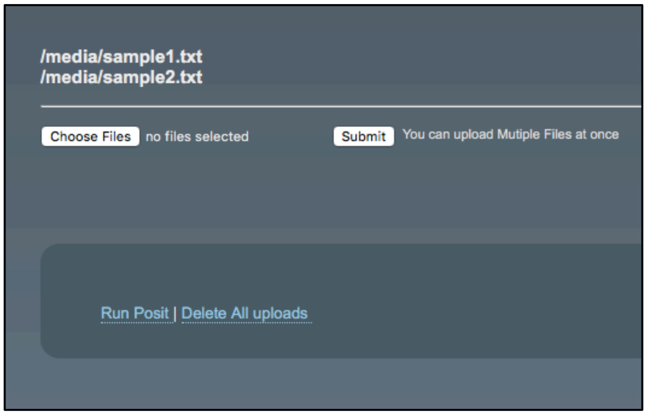
TABLE I. CLASSIFICATION RESULTS FOR DRUGS-RELATED DATA SET

Algorithm	Precision	Recall	F1
J48	0.99	0.99	0.99
kNN1	0.995	0.995	0995

V. POSIT IN THE CLOUD

In order to expand the scope and range of Posit application in textual classification tasks, we are developing a full-featured Cloud-based implementation. This facilitates third-party access to the Posit analysis of plain text data sets.

The Cloud-Posit system is being developed in four phases. In the initial Phase One version, third-parties may access an interactive Cloud-based Posit facility that affords the upload of multiple data files in a set. After file-upload, selecting the 'run Posit' option, results in Posit being executed sequentially on each file in the data set. The analysis output for each file is output as a separate folder and the complete set of analysis folders is compressed into a single file archive and downloaded to the remote Web client. Figure 4 illustrates the interactive Phase One Web interface to Cloud-Posit.



/media/sample1.txt
/media/sample2.txt

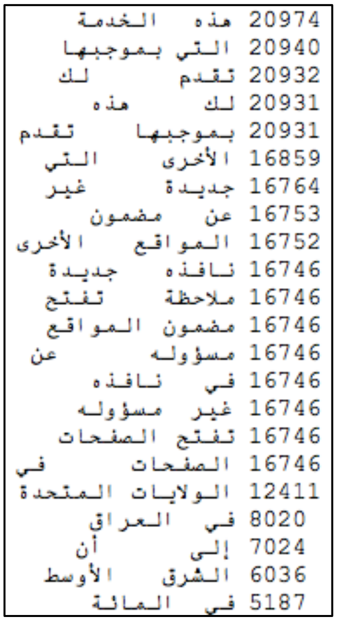
Choose Files
no files selected
Submit
You can upload Mutiple Files at once
Run Posit | Delete All uploads

Figure 4: Cloud-Posit interactive facility

This *Phase One* facility will prove convenient for initial third-party experimentation with Posit and is well-suited to small data sets. *Phase Two* of Cloud-Posit will additionally afford API access, without need for user interaction. Through this version, remote users may directly connect, upload multiple files for analysis and retrieve the result files directly, for further local processing. The expectation is that the interactive mode will be used initially by researchers seeking to 'train' their classification model, e.g., using an appropriate classification algorithm and cross-validation techniques. Once an effective model has been constructed, the bulk of analysis data would then be processed via the non-interactive API of Cloud-Posit.

The *Phase Three* version of Cloud-Posit will supplement the default Posit set of 27 features with a multiword (ngram) frequency analysis. As indicated above, ngram data is likely to provide useful additional features for use in classification. In due course, the aim is return not only raw ngram data on submitted samples, but ngram ratios for high and low frequency ngrams (e.g., against the Google ngram corpus [17]).

The planned *Phase Four* version of Cloud-Posit, will deploy parallel developments in the use of Posit for Arabic textual analysis. This applies a customized version of the Posit system and an Arabic part-of-speech tagger with output that accounts for Arabic-specific language characteristics. In addition to the standard feature set, Arabic Posit supports ngram analysis for Arabic texts. Figure 5 shows sample Arabic bigram data from Posit. For *Phase Four,* the Posit feature analysis for Arabic and the Arabic ngram analysis for 2, 3 and 4-grams will be added to Cloud-Posit.



```
20974 هذه الخدمة
20940 التي بموجبها
20932 تقدم لك
20931 لك هذه
20931 بموجبها تقدم
16859 الأخرى التي
16764 جديدة غير
16753 عن مضمون
16752 المواقع الأخرى
16746 نافذه جديدة
16746 ملاحظة تفتح
16746 مضمون المواقع
16746 مسؤوله عن
16746 في نافذه
16746 غير مسؤوله
16746 تفتح الصفحات
16746 الصفحات في
12411 الولايات المتحدة
8020 في العراق
7024 إلى أن
6036 الشرق الأوسط
5187 في المائة
```

Figure 5: Example Arabic bigrams extract from Posit

VI. Further applications

In addition to the use of Posit in supervised learning applications, there are further roles that it can play in corpus comparisons. For example, the detailed data analysis provided by Posit allows for contrastive review of two or more documents (or document sets). Such an approach was employed as a basis for gauging the similarity of grammatical approach across several generations of textbooks used to teach English in Japan [18]. We expect that the extended insights afforded by the aggregate and fine level details of Posit analysis will also find a role in classification. For this reason, our implementation of Cloud-Posit generates all three levels of quantitative textual analysis.

For any context in which quantitative analysis may shed light on textual data Posit can support such insights. By making this facility available through a Web service as Cloud-Posit, we aim to extend this utility to the academic and research community.

References.

[1] F. Sebastiani. Machine learning in automated text categorization. ACM Comput. Surv., 34(1):1–47, Mar. 2002.

[2] F. Sebastiani. Text categorization. In Encyclopedia of Database Technologies and Applications, pages 683–687. IGI Global, 2005.

[3] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A New Bench- mark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361–397, 2004. doi: 10.1145/122860.122861.

[4] B. Harish, R. M. Hegde, N. Neeti, and M. Meghana. An Empirical Study on Various Text Classifiers. Advanced Materials Research, pages 587–593, 2012. doi: 10.1109/MSR.2017.60.

[5] Y. Aphinyanaphongs, L. D. Fu, Z. Li, E. R. Peskin, E. Efstathiadis, C. F. Aliferis, and A. Statnikov. A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization. Journal of the Association for Information Science and Technology, 65(10):1964–1987, 2014. doi: 10.1002/asi.23110.

[6] V. Ha-Thuc and J.-M. Renders. Large-scale hierarchical text classification without labelled data. Proceedings of the fourth ACM international con-

ference on Web search and data mining - WSDM ’11, page 685, 2011. doi: 10.1145/1935826.1935919.

[7] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2011.

[8] G. Forman. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research, 3:1289–1305, 2003. doi: 10.1162/153244303322753670.

[9] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the 10th European Conference on Machine Learning ECML ’98, pages 137–142, 1998. doi: 10.1007/BFb0026683.

[10] F. Peng and D. Schuurmans. Combining Naive Bayes and n-Gram Language Models for Text Classification. Computer, pages 335–350, 2003. doi: 10.1007/3-540-36618-0 24.

[11] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan. Data Mining: A Knowledge Discovery Approach. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[12] G. R. S. Weir. The Posit Text Profiling Toolset. Proceedings of PAAL2007, 2007.

[13] G. R. S. Weir. Corpus profiling with the posit tools. In Proceedings of the 5th Corpus Linguistics Conference. University of Liverpool, 2009.

[14] G. R. S. Weir, E. Dos Santos, B. Cartwright, and R. Frank. Positing the problem: Enhancing classification of extremist web content through textual analysis. 2016 IEEE International Conference on Cybercrime and Computer Forensic, ICCCF 2016, pages 67–69, 2016. doi: 10.1109/ICCCF.2016.7740431.

[15] M. Bouchard, K. Joffres and R. Frank. Preliminary analytical considerations in designing a terrorism and extremism online network extractor. InComputational models of complex systems, pages 171-184, 2014, Springer, Cham.

[16] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. Artificial Intelligence, 97(1-2), pages 245–271, 1997. ISSN 00043702. doi: 10.1016/S0004-3702(97)00063-5.

[17] K. Wang, C. Thrasher, E. Viegas, X. Li and B. J. Hsu. An overview of Microsoft Web N-gram corpus and applications. InProceedings of the NAACL HLT 2010 Demonstration Session 2010 Jun 2, pages. 45-48, 2010, Association for Computational Linguistics.

[18] G. R. S. Weir, and T. Ozasa. Learning from Analysis of Japanese EFL Texts. Educational Perspectives, Journal of the College of Education/University of Hawaii at Manoa, 43 (1 & 2). pp. 56-66, 2010. ISSN 0013-1849