An Investigation Concerning the Generation of Text Summarisation Classifiers using Secondary Data

Matias Garcia-Constantino[1], Frans Coenen[1], P-J Noble[2], Alan Radford[2],
Christian Setzkorn[2] and Aine Tierney[2]

(1) Department of Computer Science, The University of Liverpool, Liverpool, L69 3BX, UK; (2) School of Veterinary Science, University of Liverpool, Leahurst, Neston, CH64 7TE, UK.
{mattgc,coenen,Rtnorle,alanrad,c.setzkorn,Aine.Tierney}@liverpool.ac.uk

Abstract. An investigation into the potential effectiveness of generating text classifiers from secondary data for the purpose of text summarisation is described. The application scenario assumes a questionnaire corpus where we wish to provide a summary regarding the nature of the free text element of such questionnaires, but no suitable training data is available. The advocated approach is to build the desired text summarisation classifiers using secondary data and then apply these classifiers, for the purpose of text summarisation, to the primary data. We refer to this approach using the acronym CGUSD (Classifier Generation Using Secondary Data). The approach is evaluated using a real questionnaire data obtained as part of the SAVSNET project.

Keywords: Text Summarisation, Text Classification Using Secondary Data

1 Introduction

Text summarisation has been a domain of research for many years. An early example (c1958) can be found in [15] in the context of literature abstracts. Within the context of data mining and machine learning researchers have looked at the application of a variety of techniques to achieve the desired summarisation, concentrating on supervised and semi-supervised learning (see for example [4]). These techniques can be further divided into statistical and linguistic techniques. The text summarisation problem is typically characterised by a large number of labels coupled with a small number of examples for each label. The requirement for supervised (and semi-supervised) learning techniques to have access to labelled *training* data is well recognised. The labelling of example documents typically requires recourse to hand annotation which in turn requires a substantial resource overhead to the extent that this approach is often rendered impractical for most applications.

This paper reports on an investigation to determine whether the desired text summarisation classifier(s) can be generated using *secondary data*, and whether the resulting classifier(s) can then be successfully applied to the *primary data* so as to produce adequately the desired summarisation (assuming, of course, that appropriate secondary data is available). We refer to this approach using the acronym CGUSD (Classifier Generation Using Secondary Data). The work is directed at a particular application, the summarisation of the free text element(s) often found in survey questionnaires. To evaluate the work a corpus of questionnaire returns from veterinary practices (where each questionnaire is concerned with a single consultation) collected as part of the SAVSNET project[1] was used. The secondary data used for evaluation purposes was the MEDLINE (Medical Literature Analysis and Retrieval System Online) life science and biomedical bibliographic database maintained by the United States Library of Medicine[2].

The rest of this paper is organised as follows. A short review of related work is presented in Section 2, and a formal definition of the problem domain is presented in Section 3. Section 4 gives a review of the proposed CGUSD process. A brief overview of the SAVSNET project is given in Section 5 together with a description of the SAVSNET veterinary questionnaire corpus. A comprehensive evaluation of CGUSD is then presented in Section 6. A summary of the main findings and some conclusions are presented in Section 7.

2 Related Work

Text summarisation techniques have been categorised in many ways according to the field of study and to other factors inherent to the text. Jones et al. [14] proposed that the key factors involved in the summarisation of text be categorised according to: (i) the input, (ii) the purpose of the summarisation and (iii) the desired output (this last category is of course related to the second). Afantenos et al. [1] provided a more "fine grained" categorisation founded on that of Jones et al. The input factors considered were: (i) the number of documents used (single-document or multi-document), (ii) the data format in which the documents are presented (text or multimedia) and (iii) the language or languages in which the text was written (monolingual, multilingual and cross-lingual). The purpose factors were sub-categorised in terms of: (i) the nature of the text (indicative or informative), (ii) how specific the summary must be for the intended audience (generic or user oriented) and (iii) how specific the summary must be in terms of the domain or field of study (general purpose or domain specific). The most significant identified output factors (amongst others) were: (i) if the summary is to be "complete", (ii) accurate and (iii) coherent. Using the categorisation proposed by Afantenos et al. the factors used in the CGUSD approach are:

- Input: multi-document, text and monolingual.
- Purpose: indicative (the relevant contents are indicated), generic and general purpose.
- Output: completeness, accuracy and coherency.

[1] http://www.liv.ac.uk/savsnet/

[2] http://www.nlm.nih.gov/databases/databases_medline.html

The "traditional" phases of text summarisation that most researchers follow are identified in [3]: (i) analysis of the input text, (ii) transformation of the input text into a form to which the adopted text summarisation technique can be applied and (iii) synthesis of the output from phase two to produce the desired summaries. For practical purposes text summarisation techniques can be divided into statistical [19, 4] and linguistic [10, 17] techniques. In this context CGUSD can be thought of as a statistical technique.

The mechanism advocated by CGUSD is to undertake the desired text summarisation using a classifier (alternative techniques found in the literature include machine learning methods founded on Naive-Bayes and Hidden Markov Models, and expert system based techniques). Text summarisation classifiers can be built using either supervised or unsupervised techniques, the distinction between the two is that the latter requires pre labelled (summarised) data. Examples of supervised techniques can be found in [7] and [5]. The technique presented in [7] uses marked segments of sentences (marked according to their significance and rethorical relations) with a set of features related to each one of them, which in turn form feature vectors that may be used by supervised learning algorithms (e.g. Naive Bayes) to extract relevant segments from the test set, thus producing the desired summarisation. The techniques presented in [5] is a semi-supervised technique based on the Classification Expectation-Maximization (CEM) unsupervised algorithm [12] (which estimates parameters of a distribution in the case of an incomplete data set) whereby a very small number of labelled examples are used to train summarisers that are based on the extraction of segments of sentences. The significance of the technique is that reported experiments indicated that it outperformed other systems that use larger amounts of labelled data. Whatever the case, the need for labelled texts, typically requiring hand labelling by domain experts, represents a costly and time-consuming overhead. Consequently, unsupervised techniques are desirable. An example of an unsupervised technique is that proposed in [16] where a graph-based ranking algorithm is applied to extract important sentences taking into account the local context of a word and information recursively produced from the entire text. Another example of an unsupervised text summerisation technique can be found in [17], where an improved version of a linear time algorithm for lexical chain computation is proposed, together with a method for evaluating lexical chains as an intermediate step. An alternative approach, and that advocated by CGUSD, is to generate a text summarisation classifier using an alternative pre-labelled data set. The use of *secondary data* to generate a classifier for text summarisation of questionnaire data, to the best knowledge of the authors, has not been previously reported in the literature.

Questionnaires typically comprise a tabular component and a free text component. A number of previous approaches to questionnaire mining have been reported in the literature. The application of data mining techniques to the tabular element of questionnaires does not present a particular challenge, tabular data mining is well understood. One example of the use of established data mining techniques directed at the tabular component of questionnaires is reported in

[6] where Fuzzy Association Rule Mining (FARM) is used to extract knowledge from the questionnaires. The mining of the free text component of questionnaires is more challenging and requires recourse to text mining techniques. An example can be found in [21] where two statistical learning techniques are used (rule analysis and correspondence analysis). The mining of complex (multi-media) data remains a focus for current research. The authors have been unable to find any reported work that uses both the tabular and the free text components of the questionnaires in the context of data mining.

3 Problem Definition

CGUSD is directed at the summarisation of questionnaire returns. The input is a collection of n questionnaires, $Q = \{q_1, q_2, \ldots, q_n\}$, where each questionnaire comprises a tabular component and a free text component, $q_i = \{Table_i, Text_i\}$ (where i is a numeric questionnaire identifier). The tabular component, in turn, comprises a subset of a global set of m attribute-value pairs $A = \{a_1, a_2, \ldots, a_m\}$; thus $Table_i \subset A$. The text element comprises sequences of words, numbers, punctuation and other printable characters. The objective is then to summarise the free text element of the questionnaires in terms of a sequence of p labels (classes), $\{c_1, c_2, \ldots, c_p\}$, where each label is drawn from k categories of labels $\{C_1, C_2, \ldots, C_k\}$, one label per category; $c_1 \in C_1$, $c_2 \in C_2$ and so on. We indicate the complete set of labels using the identifier C. The overall objective is thus to translate the input $Q = \{q_1, q_2, \ldots, q_n\}$ to a sequence of sets of labels $\{\{c_{1_1}, c_{1_2}, \ldots, c_{1_p}\}, \{c_{2_1}, c_{2_2}, \ldots, c_{2_p}\}, \ldots, \{c_{n_1}, c_{n_2}, \ldots, c_{n_p}\}\}$ such that one set of labels $\{c_{i_1}, c_{i_2}, \ldots, c_{i_p}\}$ is associated with each questionnaire q_i.

4 Classifier Generation Using Secondary Data

In this section the CGUSD approach is described in more detail. CGUSD comprises 3 stages:

1. Construction of the secondary data set.
2. Preprocessing of the secondary data.
3. Generating a classifier using the extracted and preprocessed secondary data.

The first stage involves interacting with the source from which the secondary data will be extracted, which in our case will be some kind of document collection. Recall that we wish to build k classifiers, one per category of class label. We thus require k document collections $\{D_1, D_2, \cdots, D_k\}$. We retrieve the documents required to build each classifier, utilising an “off the shelf” document retrieval system, using the individual class labels associated with the required class category C_i as the “search terms”. Thus, we conduct $|C_i|$ searches. For each search we retrieve r documents; thus, in total, for each classifier we retrieve a document collection D_i comprising $|C_i| \times r$ documents (records) (i.e $|D_i| = |C_i| \times r$). The value of r is user defined, the larger the value of r the larger the size of D_i. Experience shows that a better classifier is generated if the number of documents in D_i is such that all potential cases are "covered", and thus typically D_i needs to be of a reasonable size. However we have also discovered that in some cases, depending on the nature of the application domain, very few documents may be discovered with respect to certain search terms. If the number of documents returned is less than r we place all of the identified documents in D (running the risk, of course, that some of the documents may be only very loosely related to the search term). The reverse is also true, some search terms return a great many documents, in this case we select the "best" r (most document retrieval engines rank their results). In this manner we produce k document collections, $D = \{D_1, D_2, \cdots, D_k\}$.

In the second stage of this approach, the documents extracted from the secondary source are preprocessed in order to produce the input to be used with the selected classification software. In our case we used the widely accepted *bag-of-words* representation. Firstly, stop words (common words that are not significant for the text summarisation process) were removed from the free text (using a *stop word* list) together with punctuation. The remaining words, where appropriate, were then stemmed using an implementation of the Porter Stemming algorithm [20]. The entire document set was then recast into "lower-case". The document collection was then further processed so as to identify keywords, as a result of which each document could be represented, in terms of the identified keywords, using *feature vectors* (one per document). In the context of text mining and document retrieval, keywords may be identified in a variety of manners, however we used the well established TF-IDF (Term Frequency - Inverse Document Frequency) measure [18, 13]. TF-IDF weights are calculated for each term and the most significant terms, according to their weight, are selected.

The third stage of the proposed approach is to generate the desired classifiers, one for each category of label. There are a number of established text mining techniques that may be applied with respect to feature vector representations. With respect to the evaluation of CGUSD approach described here, the TFPC (Total From Partial Classification) algorithm [9] was adopted. This is a Classification Association Rule Mining (CARM) approach based on the Apriori-TFP (Total From Partial) Association Rule Mining (ARM) algorithm [8]. Apriori-TFP, in turn, is founded on the classic Apriori algorithm [2]. The classifiers thus generated with the secondary data may then be applied to the primary data, which must be preprocessed and prepared so that it is in a compatible format with the specific bag-of-words format identified with respect to the primary data.

Although the CGUSD approach advocates the creation of a number of classifiers, an alternative approach would be to generate a single multi-class classifier. However, some initial experiments (not reported in this paper) immediately indicated that this was not an effective mechanism.

5 The SAVSNET Application

The focus of the work described in this paper is the collection of questionnaire returns obtained as part of SAVSNET initiative[3]. SAVSNET stands for *Small Animal Veterinary Surveillance Network*, and is an initiative that is currently in progress within the Small Animal Teaching Hospital at the University of Liverpool. The objective of the SAVSNET surveillance project is to determine the disease status of small animals (mainly dogs and cats) in the UK. The project is partly supported by Vet Solutions, a software company whose software is used by 20% of the veterinary practitioners located across the UK. Some 30 veterinary practices have "signed up" to the SAVSNET initiative.

The SAVSNET veterinary questionnaires comprise a tabular (tick box) and a free text section. Each questionnaire describes a consultation and is completed by the vet conducting the consultation. The tabular section of the questionnaires is comprised of attributes relating to the consultation. Some of these attributes are concerned with the consultation (e.g. date, consultation ID, practice ID), while others are concerned with the "patient" (e.g. species, breed, sex) and its owner (e.g. postcode). The free text section of the questionnaires typically comprises notes made by the vet, which typically describe the symptoms presented, the possible diagnosis and the treatment to be prescribed. It is the free text section that we are interested in summarising, although in some cases the free text element of the questionnaires has been left "blank". In the context of the SAVSNET application we are interested in summarising the free text element of the questionnaires in terms of three categories: (i) Symptoms, (ii) Diagnosis and (iii) Treatment (thus $\{C_1, C_2, C_3\} = \{Symptoms, Diagnosis, Treatment\}$). These categories were selected in consultation with domain experts.

6 Evaluation

The evaluation of the proposed CGUSD technique, reported in this section, was directed at one particular class, namely *symptoms*. This was because, as part of the SAVSNET project, specific questions were periodically included in the tabular element of the questionnaires regarding the presence or absence of specific symptoms. These questions were included for reasons particular to another element of the SAVSNET project, but were ideally suited to supporting the evaluation of CGUSD as they provided labelled training/test sets. More specifically vets were asked to comment on the absence or presence of the following: of *"pruritus"*, *"aggression"*, *"vomiting"* and *"diarrhoea"* over a sequence of four months (one month per symptom). The research team were thus able to generate a questionnaire corpus labelled according to these symptoms. Overall the SAVSNET veterinary questionnaire corpus, used to evaluate the CGUSD approach described in this paper, comprises 944 records (out of a total of 27,072 collected records). Some statistics concerning the data set are given in Table 1. Note that all the records include tabular data, however only 828 records include free text

[3] http://www.liv.ac.uk/savsnet/

data. Note also that the data set is extremely unbalanced, only 45 records (4.76% of the total) reference *"aggression"* of which only 34 include a free text element.

	Tabular		Free Text	
Class	Num.	%	Num.	%
Aggression	45	4.76	34	4.10
Diarrhoea	336	35.59	308	37.19
Pruritus	418	44.30	350	42.27
Vomiting	145	15.35	136	16.44
Total	944	100.00	828	100.00

Table 1. Number of records per class labels

The evaluation was conducted by comparing the operation of classifiers generated using secondary data with classifiers generated from the primary data only. Three different mechanisms were considered whereby classifiers could be generated from the primary data: (i) using only the tabular data, (ii) using only the free text or (iii) using the tabular and free text data in combination.

σ	Tabular data only		Text only		Tabular and text		Secondary data on text	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	59.11	0.3698	**71.58**	**0.6394**	**66.20**	**0.4832**	18.86	0.2530
1.0	59.54	0.3712	67.95	0.5802	65.68	0.4693	16.08	0.2440
1.5	58.90	0.3610	70.14	0.5985	62.50	0.4441	16.08	0.2439
2.0	58.69	0.3546	51.65	0.4782	60.60	0.4445	15.72	0.2327
2.5	**59.75**	0.3599	45.46	0.4423	60.80	0.4700	16.20	0.2432

Table 2. Results, $\gamma = 50\%$

The required secondary dataset consisted of medical abstracts obtained from the MEDLINE database which comprises around 19 million citations for biomedical literature, including journals and books[4]. The abstracts were extracted using PubMed[5]; PubMed includes many options for searching the MEDLINE database. Each search query comprised one of the identified class labels (see Table 1) and the "English Language" and "animals" options available in PubMed. In each case r (the number of documents to be retrieved) was set to 500. Thus the final secondary data set comprises 2000 documents (500 per class label).

For the evaluation, comparisons were conducted using a range of support threshold (σ) values from 0.5 to 2.5 incremented in steps of 0.5, and a range of

[4] http://www.nlm.nih.gov/databases/databases_medline.html

[5] http://www.ncbi.nlm.nih.gov/pubmed

σ	Tabular data only		Text only		Tabular and text		Secondary data on text	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	57.43	0.3707	67.22	0.5881	65.35	0.4754	19.23	0.2516
1.0	56.03	0.3648	59.60	0.5227	65.57	0.4757	16.81	0.2451
1.5	58.48	0.3639	60.81	0.5313	62.82	0.4504	40.39	0.2393
2.0	58.48	0.3572	58.03	0.5209	61.24	0.4521	4.59	0.2466
2.5	59.32	0.3594	45.46	0.4423	60.70	0.4682	4.35	0.2450

Table 3. Results, $\gamma = 60\%$

σ	Tabular data only		Text only		Tabular and text		Secondary data on text	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	57.64	**0.3729**	67.58	0.6303	64.93	0.4797	19.83	0.2649
1.0	59.01	0.3558	62.02	0.5400	65.47	0.4663	17.65	0.2579
1.5	58.90	0.3561	53.05	0.4836	63.57	0.4563	4.96	0.2491
2.0	57.63	0.3533	50.32	0.4692	62.82	0.4668	4.59	0.2466
2.5	57.11	0.3523	45.46	0.4423	63.02	0.4893	**40.99**	0.2423

Table 4. Results, $\gamma = 70\%$

σ	Tabular data only		Text only		Tabular and text		Secondary data on text	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	52.86	0.3192	67.01	0.6001	62.50	0.4672	19.95	**0.2689**
1.0	51.70	0.3165	52.47	0.4733	57.84	0.4246	40.27	0.2453
1.5	50.00	0.3106	48.84	0.4502	52.88	0.3896	**40.99**	0.2428
2.0	46.50	0.2745	45.34	0.4339	48.01	0.3892	16.93	0.2521
2.5	47.14	0.2763	45.46	0.4423	45.24	0.3378	16.69	0.2505

Table 5. Results, $\gamma = 80\%$

confidence threshold (γ) values from 50% to 80% incremented in steps of 10%. The evaluation metrics used were overall accuracy expressed as a percentage and the Area Under the receiver operating Curve (AUC) [11]. The later was deemed to be appropriate because of the unbalanced nature of the input data. The results are presented in Tables 2, 3, 4 and 5, where each table corresponds to one of the selected confidence thresholds (γ) used: 50%, 60%, 70% and 80% respectively. Note that using primary data only the reported results were obtained using Tenfold Cross Validation (TCV). For the CGUSD approach the accuracy and AUC values were obtained as a result of applying the generated classifier to the entire primary data set. For the "Tabular data only" experiments the primary data set comprised all 944 records, while for the remaining experiments the primary data set comprised only the 828 records that included a free text element. The best accuracy and AUC values obtained with respect to each category are given in bold font in Tables 2 to 5.

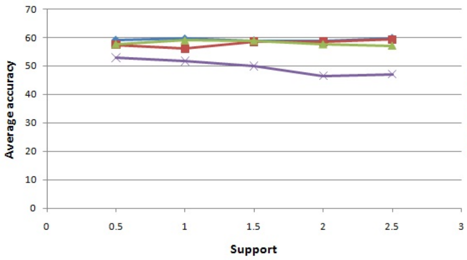

Fig. 1. Tabular data TCV (accuracy)

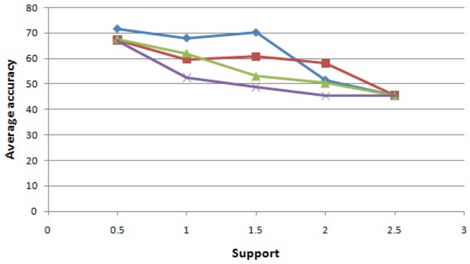

Fig. 2. Text TCV (accuracy)

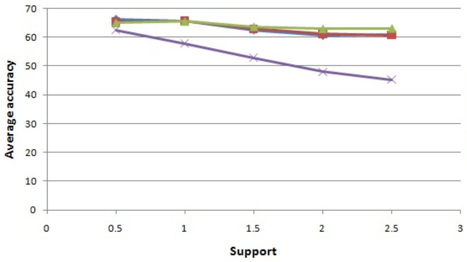

Fig. 3. Tabular and text data TCV (accuracy)

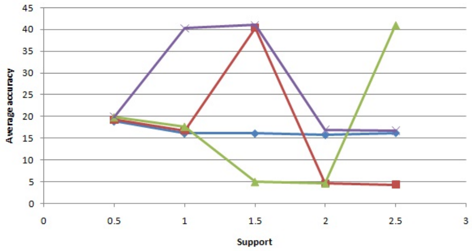

Fig. 4. Secondary data on free text (accuracy)

The same results as given in Tables 2, 3, 4 and 5 are presented in graph format in Figures 1 to 8. Figures 1 to 4 plot the obtained accuracy values against the support thresholds used with respect to the range of confidence thresholds (the

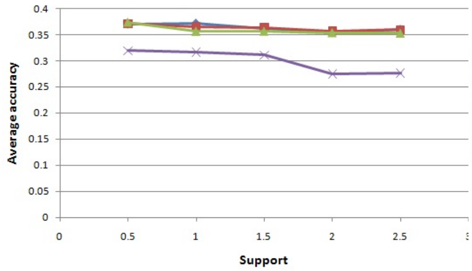

Fig. 5. Tabular data TCV (AUC)

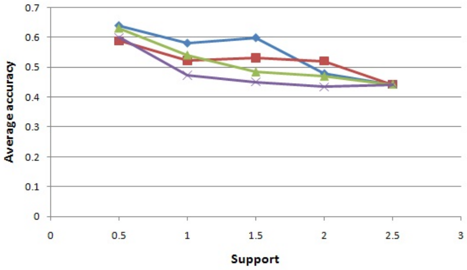

Fig. 6. Text TCV (AUC)

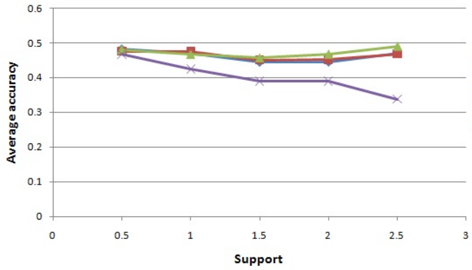

Fig. 7. Merged data TCV (AUC)

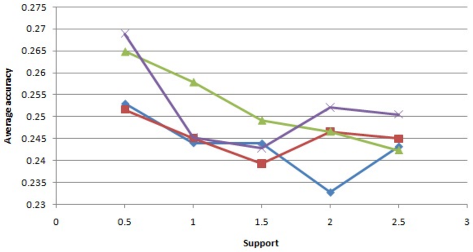

Fig. 8. Secondary data on free text (AUC)

encoding for the individual plots is given in Figure 9). Figures 5 to 8 plot the obtained AUC values against the support thresholds used with respect to the range of confidence thresholds (the same encoding for the individual plots is used as for Figure 1 to 4).

Considering the results using only primary data first; using text with a low support threshold produced the best results. Combining tabular and text data did not improve on using text only (probably as a result of over-fitting). Using tabular data on its own gave the poorest results. From this we can conclude that the processing of the free text element of the questionnaires is important in the context of the desired summarisation. The best accuracy obtained using primary data was in the order of 72%.

When looking at the results produced using the secondary data, the first observation (and to some extent the most obvious) is that use of secondary data does not produce as good results as when using primary data (assuming of course that such data is available). The best accuracy obtained is 41%. With reference to Table 1 we can note that by simply classifying everything as “Pruritus” we can achieve an accuracy of 42%. However, working under the assumption that we do not know the class priors and thus assuming an equal distribution (as would be the cases where we have only unlabeled data) than anything better than 25% would be good. Thus we can conclude that using secondary data to build the desired classifier, although not better than in the case when primary data can be used, does provide a result that is better than a "guess". We can identify a number of reasons why the operation of the CGUSD approach may not have been as effective as expected, as follows.

1. **Compatibility between secondary training set and primary test set.** The quality of any supervised learning method is very much dependent on the quality of the input. In the case of the evaluation, MEDLINE abstracts were used. The purpose of abstracts (a brief summary of a research article designed to inform potential readers) is very different to the notes found in the free text of SAVSNET questionnaires. It may therefore be that there may have only been a limited compatibility between the secondary training data set and the primary test data set and thus the generated classifier would not have been precisely suited to classifying examples contained in the primary data.
2. **Class priors.** As noted above the primary data was unbalanced, whilst the secondary data was balanced. Knowledge of the class priors, reflected in the classifiers generated directly from the primary data, would produce a better classifier. However, as also noted above, then assumption underpinning CGUSD is that this knowledge is not available.
3. **Evaluation environment.** The evaluation environment, as described above, compares accuracy and AUC values obtained using two different mechanisms, TCV and application to an entire data set. The first obvious distinction is that the relative size of the test sets are very different (by a ration of ten to one). Secondly the quoted values using TCV are the result of an averaging process. It may therefore be conjectured that we are not comparing like-with-like; although it is difficult to see any reasonable alternative.

7 Conclusion

This paper reports on experiments conducted to ascertain the effectiveness of a process whereby a classifier is generated using an alternative data source to that for which it is intended. The application scenario where this is seen as applicable is where we wish to build and apply a classifier to data, but have no labelled data with which to "train" the classifier. More specifically where we wish to build text summarisation classifiers, that can be applied to the free text element of questionnaire data, where no labelled training data is available. The advocated approach is therefore to build the classifier using an alternative secondary data source which is labelled and then apply this to the primary data source. An approach, which we have called CGUSD, is therefore proposed whereby this can be achieved. The focus of the work described is text summarisation and CGUSD assumes that this is the intended objective.

The operation of CGUSD was tested using questionnaire data obtained as part of the SAVSNET project. The secondary data set was generated using MEDLINE abstracts. The results obtained, although not as good as was hoped for, indicate that (in the absence of any alternative) CGUSD does present a potential solution; although the technique does require further refinement. (Clearly we could not expect to produce a better result than when using primary data.) However, the research team has been encouraged by the results that have been produced to date.

References

1. Afantenos, S. and Karkaletsis, V. and Stamatopoulos, P. (2005). Summarization from medical documents: a survey. Artificial Intelligence in Medicine Vol. 33, pp157-177.
2. Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Data Bases.
3. Alonso, L. and Castellón, I. and Climent, S. and Fuentes, M. and Padró, L. and Rodríguez, H. (2004). Approaches to text summarization: Questions and answers. Inteligencia Artificial Vol. 8, pp22.
4. Amini, M-R. and Gallinari, P. (2001). Automatic Text Summarisation Using Unsupervised and Semi-Supervised Learning. Proc. PKDD 2001. Springer LNAI 2168, pp16-28.
5. Amini, M-R. and Gallinari, P. (2002). The use of unlabeled data to improve supervised learning for text summarization. SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.
6. Chen, Y.L. and Weng, C.H. (2009). Mining fuzzy association rules from questionnaire data. Knowledge-Based Systems Vol. 22, pp46-56.
7. Chuang, W.T. and Yang, J. (2000). Extracting sentence segments for text summarization: a machine learning approach. SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp152-159.
8. Coenen, F. (2004). The LUCS-KDD TFP Association Rule Mining Algorithm. `http://www.csc.liv.ac.uk/~frans/KDD/Software/Apriori_TFP/aprioriTFP.html` Department of Computer Science, The University of Liverpool, UK.
9. Coenen, F. (2004). The LUCS-KDD TFPC Classification Association Rule Mining Algorithm. `http://www.cSc.liv.ac.uk/~frans/KDD/Software/Apriori_TFPC/aprioriTFPC.html` Department of Computer Science, The University of Liverpool, UK.
10. Fuentes, M. and Rodríguez, H. (2002). Using cohesive properties of text for automatic summarization. JOTRI02.
11. Hand, D.J. and Till, R.J. (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classication Problems. Machine Learning, 45, pp171186.
12. Jebara, T. and Pentland, A. (1999). Maximum conditional likelihood via bound maximization and the CEM algorithm. Advances in neural information processing systems, pp494-500.
13. Jing, L.P. and Huang, H.K. and Shi, H.B. (2002). Improved feature selection approach TFIDF in text mining. Proceedings of the First International Conference on Machine Learning and Cybernetics.
14. Jones, K.S. and others. (1999). Automatic summarizing: factors and directions. Advances in automatic text summarization, pp1-12.
15. Luhn, P.H. (1958). Automatic creation of Literature Abstracts. IBM Journal, pp159-165.

16. Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL 2004)(companion volume).
17. Silber, H.G. and McCoy, K.F. (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. Computational Linguistics Vol. 28, pp487-496.
18. Sparck Jones, Karen (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28 (1): pp1121,
19. Strzalkowski, T. and Wang, J. and Wise, B. (1999). A robust practical text summarization. Proceedings of the AAAI Symposium on Intelligent Text Summarization.
20. Willett, P. (2006). The Porter stemming algorithm: then and now. Program: electronic library and information systems Vol. 40, pp219-223.
21. Yamanishi, K. and Li, H. (2002). Mining open answers in questionnaire data. IEEE Intelligent Systems, pp58-63.