Yuha Yi
Stats 517
Stephen Lee

Machine Learning in Automated Text Categorization by Fabrizio Sebastiani

The usages of text categorization have been explored within the past few years to streamline businesses, collect knowledge more efficiently, and store information in a way that can be used to our advantage. Such usage of information is now digitally stored by users nearly every day from online sources of social networks, e-commerce markets, and feedback ratings. These sources are critical in gathering knowledge to forecasting businesses, creating an efficient market supply, and improving target audiences for more customers. The metadata from images, user accounts, and documents can be automatically processed to help businesses. Machine learning (ML) has the ability to classify bodies of text as information from a set of pre-classified documents.

Text classification can be described within a few steps. The first step being dataset preparation. This includes observing the data, getting rid of superfluous words, and splitting it into a train and test set. The second step is changing the data into different features. This makes the dataset easier to read for the ML algorithm to classify into easier groups. Once the features have been identified, it is critical to obtain which features are most important and in what ways can dimensionality reduction can be used to improve the model. The last step is finding a proper and most improvised text classifier model. This is the most difficult step and takes intuition in different machine learning processes and statistical learning.

Text classification can be described as the task of building text classifiers, i.e. software systems that classify documents from a domain D into a given, fixed set C = {c_1, …, c_m} of pre-defined categories (classes or labels). An example of pre-defined categories in a document (or set of words) is a newspaper. The newspaper is organized by subject categories known as topics that may include weather, crimes, local news, and national news. In text classification, it is assumed that a ground truth exists such that it specifies how a document should be classified. However, because it isn’t known how it should be classified, text classification consists in building a method that approximates this classification. Text classification uses complex algorithms to transform text into certain categories to values. The first step is to categorize these words into Boolean categories. Depending on the category selection, the dataset can be organized by its labels in two different ways, single label and multilabel text categorization. The difference being such that the multilabel can have overlapping categories. In detail, single-label text classification is when exactly one category must be assigned to each document. Multi-label can have any number of categories assigned to each document. This is also known as apply a target function.

Document filtering is used within the process of delivering the proper documents to the correct audience. An example of such is described as someone being interested in all of the sports magazines and someone who is not. Such a filter would be able to identify the magazines using a single classification (Boolean) method by separating them as sports and non-sports. A more modern usage of document filtering is the spam folder in an email being classified as either spam or non-spam. A ML method would be able to use key words that often appear in spam mail to classify it as spam. Such words may consist of “Wow”, “Winner” or “Free cruise to the Bahamas”. An example of multi class labelling is figuring out whether a body of text is a question, inquiry, bad review, or a good review. A customer may post on a business about a product with the following text “This product was great!” while another customer may comment “I cannot find this item. Do you know where it is?”. The classifiers would be able to obtain key words such as “good”, “great”, and “bad” to identify whether the comment was a review. From those given words it would then identify whether it was a good review or a bad one. The second comment shows identifying key words or phrases that imply a question such as “Do you know” or “where”. The separation of identifying the cluster of words as a review, question, inquiry or some other classifies this case as a multi labeled scenario.

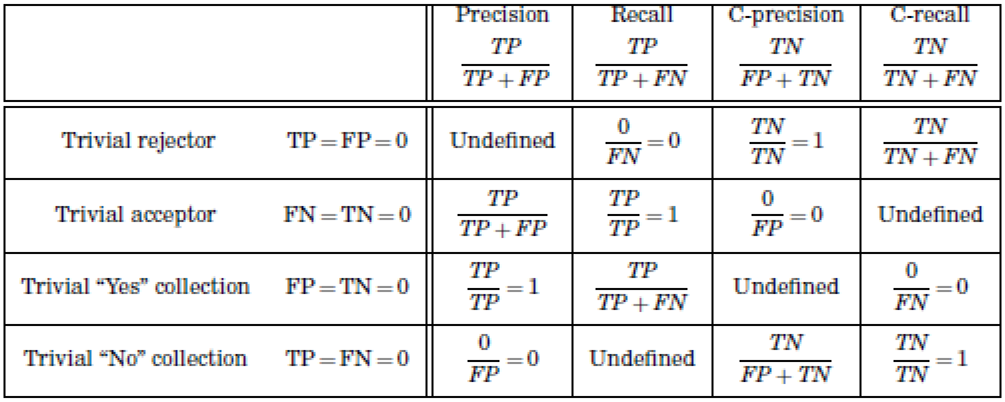
One issue that occurs within the process of text classification can be described as "word sense disambiguation". This is the idea that several words have two different meanings. The word "fair" can be applied in going to the book fair, or a fair decision that was made. Words around the ambiguity are then formed together to provide a decision to provide the proper meaning intended, this process is called a rule-based classifier. This is an important concept called *bag of words* that will be described later in the paper. To help with other forms of noise, text cleaning can be done to remove common words that have no meaning. Common words that are removed are "the", "and" and "at".

Machine learning classification can be applied in two different ways in text classification. Document-pivoted classification (DPC) is suitable when documents become available one at a time (e.g. in e-mail filtering). Category-pivoted classification (CPC) is suitable when new categories may be added after a number of documents have already been classified (e.g. in patent classification). The paper focuses on DPC because it is the most common in text classifier applications. The supervised learning approach utilizes a train-test set to a set of documents previously classified called a labelled corpus, denoted as Ω. The labelled corpus thus constitutes a glimpse of the ground truth from each known pair of the training-testing set where $\Omega = \{d1, \dots , d_{|\Omega|}\}$ where d_x are all subsets in of classified documents in domain D. The training set is built by observing the characteristics of the documents inductively. The testing set is fed to the classifier and its decisions are to be compared. The effectiveness is based on how often the decisions it made match the training set. Often times the classifier is retrained on the initial corpus to boost effectiveness. The second ML method the paper covers is the k-fold-cross validation approach; this is especially used when the training corpus is small.

The text classification process consists of three phases, document indexing, classifier learning, and classifier evaluation. Document indexing is the process of associating or tagging documents with different "search" terms. Indexing creates the "searchable" information that can be used later to easily find and identify documents. Examples can be certain dates, customer/employee number, invoice number, and other important identifies in text. The information can be stored or integrated into a database or records management system. There are two types of indexing. Full-text indexing implies its own name; all the text of the document is indexed. When specific words or descriptions are indexed to create the searchable index fields, the information is referred to as "metadata". This method is a form of information retrieval. The bags of words method simplifies information retrieval by counting frequency of how many times a word appears within a document, stores it in a vector, and gives it a weight. One issue with term frequency is the appearance of words that give no meaning such as the words "the", "a", "to" while the more meaningful terms that do not appear as often are not accounted for. To address this problem, one of the most popular method used is called term frequency-inverse document frequency. As it suggests, an inverse document frequency factor is used which diminishes the weight of words that occur too often and increases the weight of terms that occur rarely. As previously mentioned above, word sense disambiguation can occur with the bag of words methods such that two words that are spelled the same way would be counted as the same word. A grouping of words is used to make sense of the words around it. In this approach, called N-grams, each word is called a gram. Depending on the number of grams you select, it creates a vocabulary of words of N groupings. For example, if N is at two, then a sentence of "How are you doing" would be grouped as "how are", "are you", "you doing". This allows the count of how the words are being used together, if there is a noun next to a verb or other such information. Classifier learning takes the training sets and outputs a classifier. The most common ML technique for classifier learning is linear classification.

Once the machine learning process is complete, it is then scored on it's performance. This next process entails the measures of text categorization effectiveness. Classification effectiveness is usually measured in terms of precision and recall. Precision is the number of true positives over the number of true positives plus false positives. Recall is the number of true positives over the number true positives plus false negatives. A table of the listed measurements of efficiency is below.

Figure 1

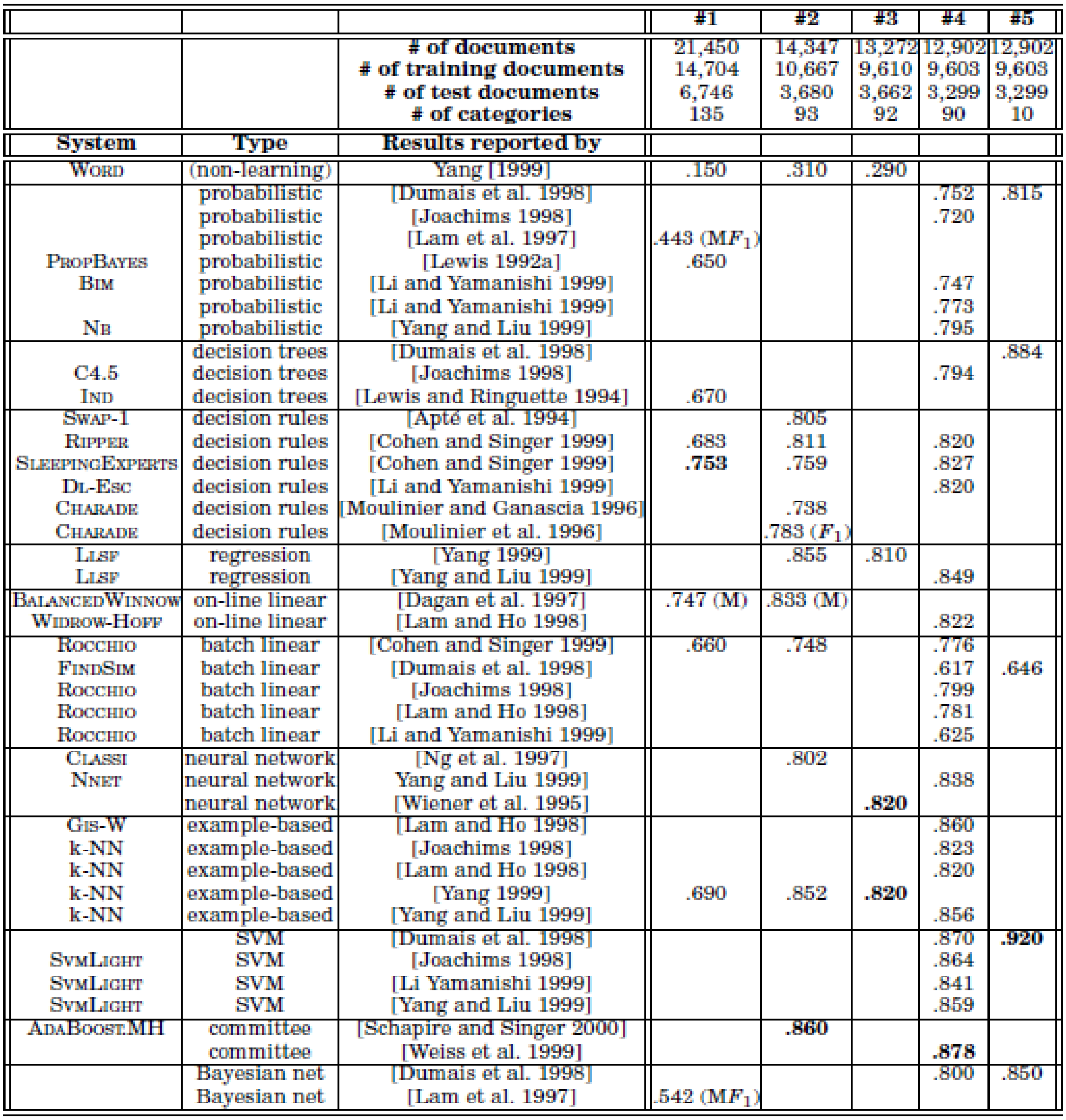


		Precision $\frac{TP}{TP+FP}$	Recall $\frac{TP}{TP+FN}$	C-precision $\frac{TN}{FP+TN}$	C-recall $\frac{TN}{TN+FN}$
Trivial rejector	TP = FP = 0	Undefined	$\frac{0}{FN}=0$	$\frac{TN}{TN}=1$	$\frac{TN}{TN+FN}$
Trivial acceptor	FN = TN = 0	$\frac{TP}{TP+FP}$	$\frac{TP}{TP}=1$	$\frac{0}{FP}=0$	Undefined
Trivial "Yes" collection	FP = TN = 0	$\frac{TP}{TP}=1$	$\frac{TP}{TP+FN}$	Undefined	$\frac{0}{FN}=0$
Trivial "No" collection	TP = FN = 0	$\frac{0}{FP}=0$	Undefined	$\frac{TN}{FP+TN}$	$\frac{TN}{TN}=1$

TP = True Positives FP = False Positives   TN = True Negatives FN False Negatives

The model that was observed to be the best appeared to be the light support vector machine, and the ada boost multi-class hamming trees method. Although it may be the most accurate, the runtime algorithms, resources, and other variables take play in selecting the most optimal model. It is important to test against multiple models to maximize efficiency. Different techniques can be used to blend multiple models together and further improve results. This can help improve the accuracy and have better implementation into a model. A summary of the observed classifications can be seen in figure 2.

The paper is heavily technical in the mathematical and statistical aspects which makes it difficult for those without a mathematics background difficult to understand. Sebastiani does well in explaining the theory behind the equations and covers all areas of text classification. The goal for this literature review is to allow a better understanding of text categorization for those not heavily involved in pure mathematics, but still have an understanding of a basic statistical background. Text categorization has caught the eye of many businesses within the last few years due to the digitalization of our era. Text categorization is a difficult process to achieve however, results in an automated process that can save thousands of hours and process an amount of data no human can accomplish within the short amount of time a machine learning algorithm can achieve.



			#1	#2	#3	#4	#5
		# of documents	21,450	14,347	13,272	12,902	12,902
		# of training documents	14,704	10,667	9,610	9,603	9,603
		# of test documents	6,746	3,680	3,662	3,299	3,299
		# of categories	135	93	92	90	10
System	**Type**	**Results reported by**					
Word	(non-learning)	Yang [1999]	.150	.310	.290		
	probabilistic	[Dumais et al. 1998]				.752	.815
	probabilistic	[Joachims 1998]				.720	
	probabilistic	[Lam et al. 1997]	.443 (MF_1)				
PropBayes	probabilistic	[Lewis 1992a]	.650				
Bim	probabilistic	[Li and Yamanishi 1999]				.747	
	probabilistic	[Li and Yamanishi 1999]				.773	
Nb	probabilistic	[Yang and Liu 1999]				.795	
	decision trees	[Dumais et al. 1998]					.884
C4.5	decision trees	[Joachims 1998]				.794	
Ind	decision trees	[Lewis and Ringuette 1994]	.670				
Swap-1	decision rules	[Apté et al. 1994]		.805			
Ripper	decision rules	[Cohen and Singer 1999]	.683	.811		.820	
SleepingExperts	decision rules	[Cohen and Singer 1999]	**.753**	.759		.827	
Dl-Esc	decision rules	[Li and Yamanishi 1999]				.820	
Charade	decision rules	[Moulinier and Ganascia 1996]		.738			
Charade	decision rules	[Moulinier et al. 1996]		.783 (F_1)			
Lisf	regression	[Yang 1999]		.855	.810		
Lisf	regression	[Yang and Liu 1999]				.849	
BalancedWinnow	on-line linear	[Dagan et al. 1997]	.747 (M)	.833 (M)			
Widrow-Hoff	on-line linear	[Lam and Ho 1998]				.822	
Rocchio	batch linear	[Cohen and Singer 1999]	.660	.748		.776	
FindSim	batch linear	[Dumais et al. 1998]				.617	.646
Rocchio	batch linear	[Joachims 1998]				.799	
Rocchio	batch linear	[Lam and Ho 1998]				.781	
Rocchio	batch linear	[Li and Yamanishi 1999]				.625	
Classi	neural network	[Ng et al. 1997]		.802			
Nnet	neural network	Yang and Liu 1999]				.838	
	neural network	[Wiener et al. 1995]			**.820**		
Gis-W	example-based	[Lam and Ho 1998]				.860	
k-NN	example-based	[Joachims 1998]				.823	
k-NN	example-based	[Lam and Ho 1998]				.820	
k-NN	example-based	[Yang 1999]	.690	.852	**.820**		
k-NN	example-based	[Yang and Liu 1999]				.856	
	SVM	[Dumais et al. 1998]				.870	**.920**
SvmLight	SVM	[Joachims 1998]				.864	
SvmLight	SVM	[Li Yamanishi 1999]				.841	
SvmLight	SVM	[Yang and Liu 1999]				.859	
AdaBoost.MH	committee	[Schapire and Singer 2000]		**.860**			
	committee	[Weiss et al. 1999]				**.878**	
	Bayesian net	[Dumais et al. 1998]				.800	.850
	Bayesian net	[Lam et al. 1997]	.542 (MF_1)				

Figure 2 The results observed by others in text categorization with the scoring and their respective classifier