Text classification : technologies and application

Autor(en): **Burschka, Stefan / Reitmann, Marcel / Varone, Sacha**

Objekttyp: **Article**

Zeitschrift: **Comtec : Informations- und Telekommunikationstechnologie = information and telecommunication technology**

Band (Jahr): **79 (2001)**

Heft 9

PDF erstellt am: **11.07.2024**

Persistenter Link: https://doi.org/10.5169/seals-876565

Text Classification: Technologies and Applications

For a service provider like Swisscom, one of the most important promises of automated text classification is that it will help to reduce the manual workload of classifying electronic texts and therefore help to accelerate the response time when dealing with customer requests. There are, however, various factors to be considered when introducing classification engines into customer relationship management processes. For example, pre-processing the data by means of statistical analysis can greatly improve the classification results. Furthermore, knowing the characteristics of the different classification algorithms helps choosing the right packages for the problem at hand.

The programme "The Net-Centric Application Business" explores the opportunity for remote applications and application service providing models that result from the expected availability of broadband Internet access, both fixed and mobile, and the evolution of various end-devices for residential and business customers. With its Exploration Programmes, Corporate Technology is exploring telecommunication technologies and new service possibilities with a long-term view of 2–5 years. Further, the expertise built up in the course of this activity enables active support of business innovation projects.

STEFAN BURSCHKA,
MARCEL REITMANN, SACHA VARONE
AND FREDERIC DREIER

The purpose to use automatic text classification is mainly to decrease the time spent on manually classifying new documents. Given the large amount of text (documents, web pages, e-mails etc.) produced today, an individual is completely overtaxed to filter out the information and data relevant to him, to some other person or to a certain context.
Applications for automated text classification are almost unlimited. A few important application fields are given below.

Ordering and Fulfilment Processes of a Service Provider

Unstructured orders (plain text) sent by e-mail, letter, fax etc. are automatically classified and routed to the corresponding person or fulfilment process. Of course, incoming letters and faxes on paper would have to be digitalised first.

Knowledge Management Processes of a Company

Automated text classification can efficiently support knowledge management processes. When a user produces a new document or receives an unclassified document, a classification engine would propose to him the category of the document management system where it fits in best. The aim of classifying documents is finally to be able to quickly retrieve information and knowledge on a specific topic: Business-Unit-wide, Corporate-wide or even world-wide (search).
Example: Someone wants to find all information available about billing of GPRS services. The document management system has to be able to retrieve this information quickly, without scanning all documents available on the servers, and accurately, without giving irrelevant information e.g. about the technical specification of GPRS. To achieve such a successful search, an engine has to correctly classify all available documents.
Automated text classification can also be used for filtering relevant information out of the daily information flood (e.g. e-mail) according to an individual profile. In this context users might also subscribe to library services to be permanently provided with new documents concerning specific topics.

Content Classification

Widening out the application field of automated text classification we come to automated content classification. Automated content classification will not only allow to classify texts, documents, e-mails etc. but also objects like films, videos, articles in a store etc. Text classification engines can be used for classifying content in general, in case a descriptive text (also referred to as "meta text") on the objects is available. This meta text has to describe as accurately as possible the features of the objects to classify. For example, if we would like to be able to quickly find all movies where the actor Gary Cooper appears, we would have to classify all movie descriptions available in the descriptive text database by the feature "Gary Cooper". In case there is no descriptive text available it would nevertheless be possible to find the movies with Gary Cooper e.g. by using pattern recognition methods. Using this method the engine would have to be taught first by a human which pattern matches with the feature "Gary Cooper".

Analytical CRM

Another application field could be customer behaviour analysis being a part of the Customer Relationship Management (CRM) process: Tracking click streams and transactions (Web browsing, using e-Services, ordering, etc.) of Web users it may be possible to categorise Web customers by certain patterns. These user categories could then be used for example for direct marketing purposes.

As automated text classification engines may be based on a variety of algorithms which have different performance characteristics, the successful application of an engine to a specific domain greatly depends on insight knowledge about the algorithms as well as collateral measures on the classification material itself.

Supervised and Unsupervised Classification

For classifying texts there are two general approaches: Supervised and unsupervised methods. The supervised meth-

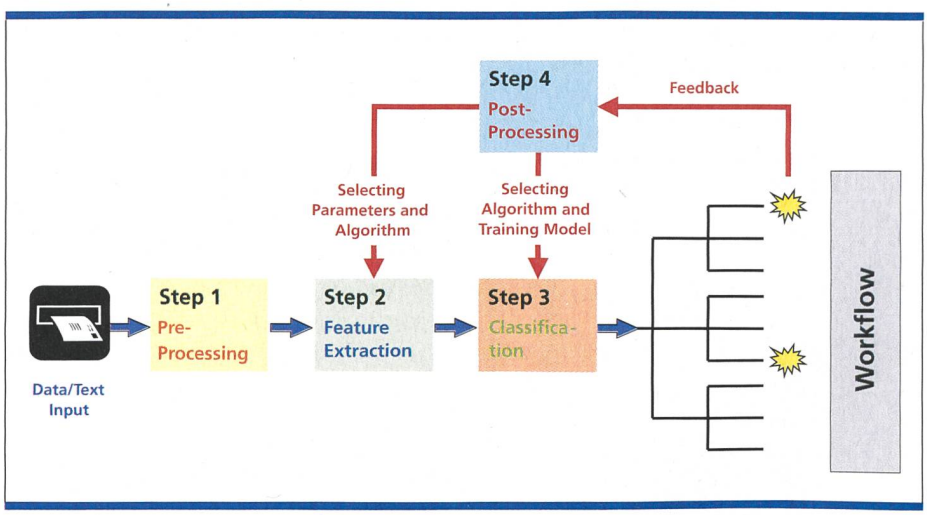

Fig. 1. The four steps of the automated Text Classification process.

ods establish classification rules from a given pre-classified data set. This means that the categories of classification have already been defined and contain manually classified texts (by a human). New texts have to be classified based on these manually classified texts. As a consequence someone always has to define the categories in advance and to do some manual classification. Unsupervised methods, also called clustering methods, will support this process: Unsupervised methods automatically propose a classification based on similarities among the available text samples. In this article, supervised text classification is considered exclusively.

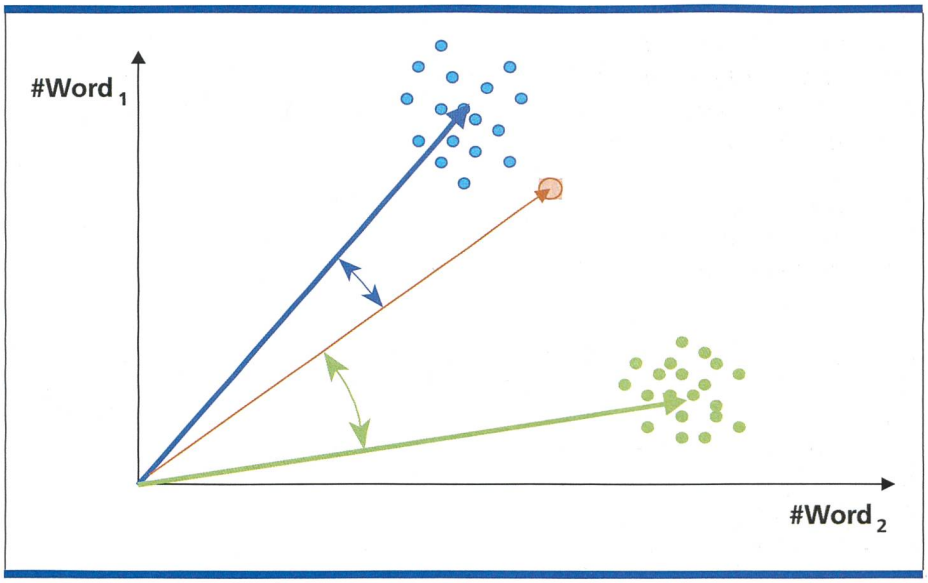

Fig. 2. Principal function of a classification engine, for simplicity with two dimensional input space #Word$_1$ and #Word$_2$.

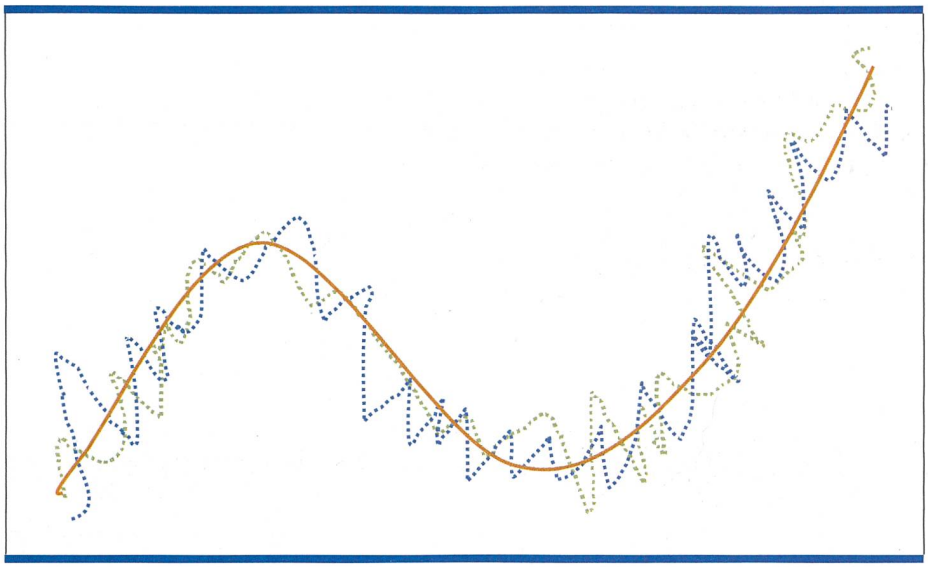

Fig. 3. The process of model building explained as a function regression task: The generalised model (red curve) acquired from the training data (blue curve) represents the important qualities in the test data (green curve). Good Generalisation Performance.

Process of Automated Text Classification

The complete process of automated text classification can be separated into 4 steps which are depicted in figure 1 and described in detail below.

Step 1: Pre-processing

During pre-processing a proper coding or representation of the training text as the input for the classification engine has to be chosen. Ignoring certain ingredients in a text, e.g. attachments or pictures, should simplify the representation of the input stream for the classification engine. The resulting easiest representation could be a vector consisting of the frequency of words (frequency of word k denominated as #Word$_k$) occurring in all training texts, as shown below:
(#Word$_1$, #Word$_2$...)
Here the vector dimension is the total number of different words available in all training documents. So "#Word$_i$" is called a metric which enables us to measure the difference between certain texts in order to classify them. This difference could be expressed e.g. in a vector difference or an angle between the vectors.
There is more than one metric possible, e.g. the frequency of ASCII characters itself, or a combination of words. Also, linguistic information could be part of the metric. Nevertheless, if a metric does not supply enough degree of freedom for distinguishing the texts into classes, the resulting classification output will be useless.
Often statistical analysis is necessary in order to simplify and expedite the training and classification process. The resulting linear or non-linear transformations on the input vector will then be used in the so-called Feature Extraction.

Step 2: Feature Extraction

A feature is a word or group of words directly used by the classification algorithm to classify pieces of plain text into predefined categories. The idea of Feature Extraction is to find significant features which best describe the different classes and supply a good measure in order to classify new unclassified text samples. Irrelevant features are also called "stop words". What a significant feature is depends on the type of classification problem. Stop words in one classification process could be significant features in another one.
Feature Extraction is an algorithm used to extract features from plain text or more generally from document content. Features from plain text objects are usually keywords or combinations of keywords. For text based feature extraction statistical methods are used. Also algorithms reducing the words to their roots ("stemming") belong to feature extraction, reducing the amount of irrelevant input data. Feature Extraction, if done correctly, leads to a better performance of the classification engine.

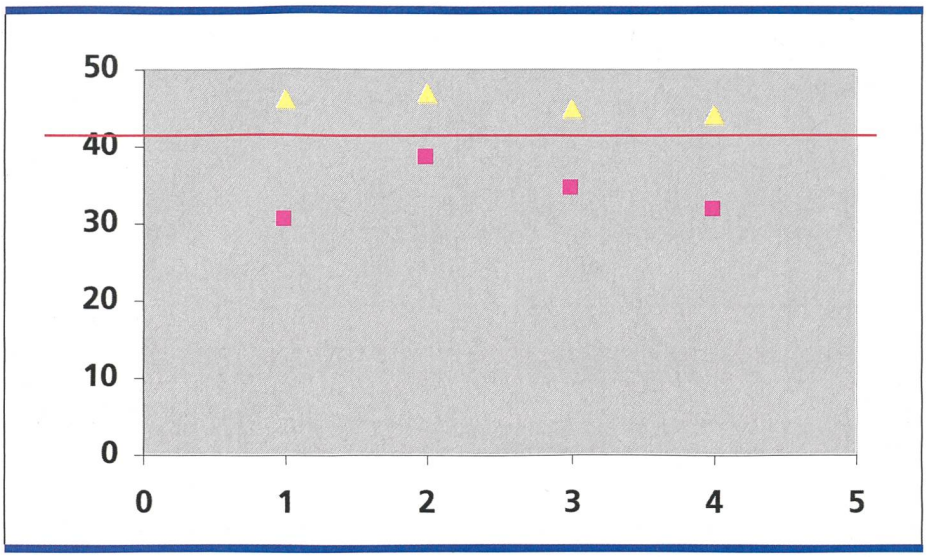

Fig. 4. Example of a linear regression: The triangles all belong to the Triangle category and all the squares belong to the Square category. A straight line separating the two categories was found with a linear regression method.

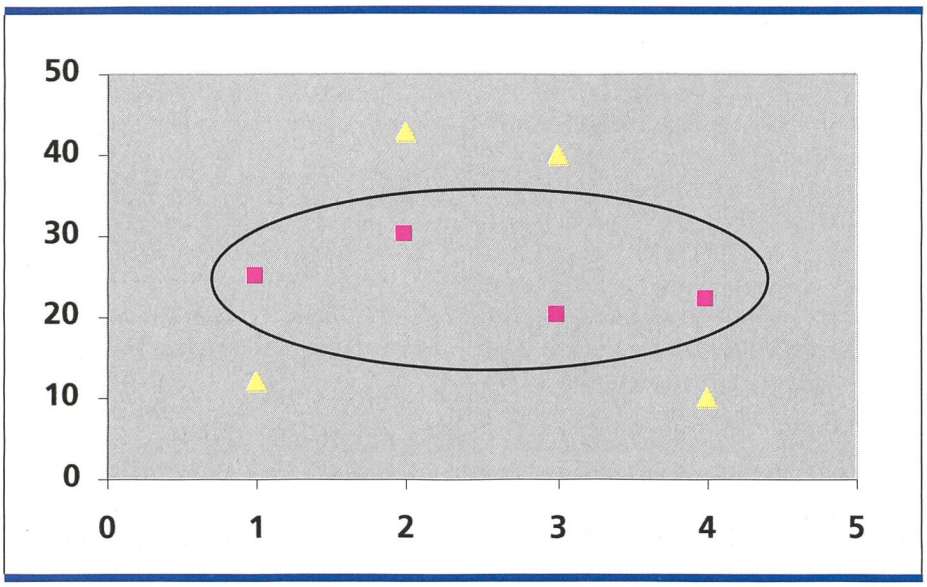

Fig. 5. Example of a quadratic discriminant: The boundary is not a straight line as in the linear case, but a quadratic curve.

Step 3: Classification

During the classification process the features extracted before are automatically matched with the pre-defined categories. The classification methods used (the most important ones are described in more detail below) are very different, but they all have in common that the category decision is made based on the "historical" data available at the moment, i.e. all already "correctly" classified or labelled texts. Classification algorithms will use this data as training set to learn from. Of course the basis of the training set has to be large enough and representative of the classes as illustrated in figure 2. Here the colour of the circles represent the training sets of the two classes "green" and "blue". The classification algorithm should learn the representative centre, drawn as a bold vector. New data, the red one, will be classified for example by an angular measurement in our selected metric, described under Step 1. Therefore, the new data point will be assigned to the blue class having the smallest distance to its representing centre vector.

Another important characteristic of classification algorithms is their ability to generalise or to ignore irrelevant details during the training process as illustrated in figure 3. The blue curve represents the training data. If all details of its outline had been modelled, a green test curve would not have been recognised as a class member. Thus the so-called Generalisation Performance is poor. If the red curve had been the element of the resulting model, all noisy data would have been excluded, resulting in a correct "low pass behaviour" now filtering the red curve behaviour in all test data.

Step 4: Post-processing

Having a text classification engine in operation it will be necessary to periodically supervise the achieved results. It has to be checked whether the engine works sufficiently accurate and whether sufficient samples were classified at all. Here the Average Probability of Misclassification (APM) provides a measure whether the results are acceptable or not. In case of high or rising APM the training set or the categories would have to be updated. Either on-line during normal operation or off-line in a batch mode.

Subsequent Workflow

Considering ordering and fulfilment processes of a service provider, data classified once (orders) can be used to trigger and control the subsequent fulfilment workflow. This workflow can be automated as well. In this case it would be stored as workflow rules in an additional knowledge base.

Text Classification Algorithms

Automatic text classification (or categorisation) is defined by an assignment of category labels to new documents based on the likelihood suggested by a training set of labelled documents.

Regression Methods

Regression methods are based on finding the boundaries of the categories. A point in a particular space represents a document. The aim is to find separators that can delimit the region of the space covered by a category. Linear regression (fig. 4), logistic discriminant, quadratic discriminant (fig. 5) or Support Vector Machines (SVM) belong to the class of regression methods.

K-nearest Neighbours

The principle of the k-nearest neighbours method is to classify a new item x according to the category containing most of its k-nearest neighbours (fig. 6).

Naïve Bayes or Bayesian Networks

Naïve Bayes is based on the calculus of probabilities. The principle is to consider the probability of a particular data (document), given a model. A Bayesian network is a directed acyclic graph (directed tree graph) of nodes representing random variables and links representing probabilistic dependencies. Absence of links would mean conditional independence among nodes (fig. 7). The Naïve Bayesian approach with a machine learning and a manual refinement algorithm is for example used by the commercial CRM tool "Kana Classify".

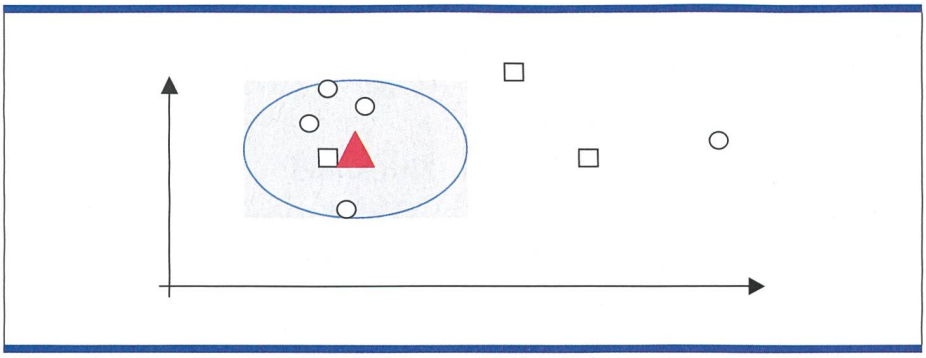

Fig. 6. Example of a k-nearest neighbour: The new item triangle is surrounded by a disc which contains four elements of class Circle, and one element of class Square. Therefore the new item will be classified in the class Circle.

Machine Learning of Rules

This method is based on recursive partitioning of the attribute space. The space is divided into boxes, and at each stage in the procedure each box is examined to see if it may be split into two boxes.
Example of a rule: Suppose a document d has to be classified. The words "GPRS", "Technical", "Price" and "Cost" exist in the training set of documents. If d contains the words "GPRS" and ("Price" or "Cost") and not "Technical", then d belongs to category A.
For example the commercial CRM tool "e-Gain AI" applies such a method:
A phrase vector for each incoming message is compared with the phrase vectors for categories in a knowledge database. Additionally, this e-gain uses a semantic layer and a syntactical layer. The syntactical layer examines the structure of the sentence by checking the sequence of the words. The semantic layer uses an algorithm to reduce the words to their roots (stemming).

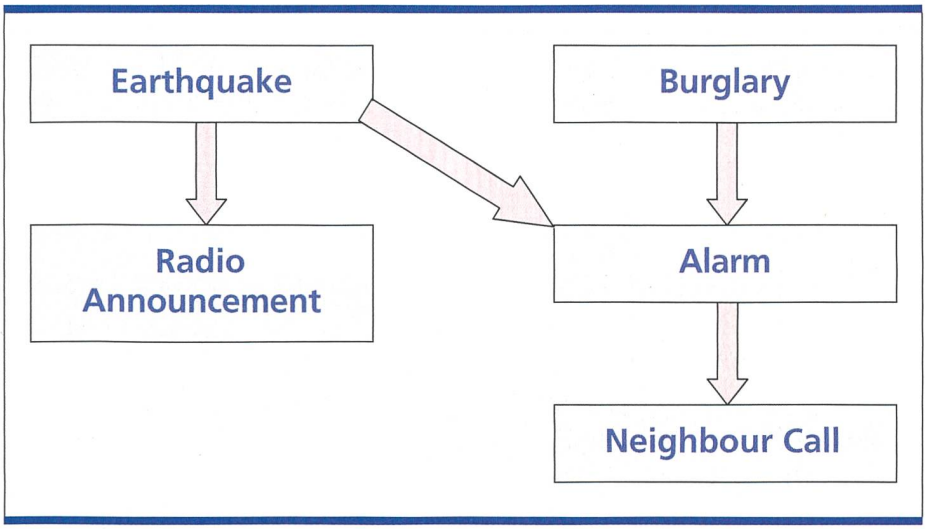

Fig. 7. Example of a Bayesian Network: Probabilities on each node are associated with this directed graph model. This means that, for example, given the state of Earthquake and Burglary, the probability that an alarm occurs is known.

Neural Networks

Neural networks are processes that enable to model highly non-linear functions. Basically, they can be represented as a function: given an input, they return an output. Neural networks in their beginning aimed at reproducing mathematically the functionality of a human brain. The representation is a directed weighted graph with functions associated to some of its nodes (fig. 8).

Results of Text Classification

The methods for classifying text documents are not all equally efficient on the same data. Moreover, one method could be the most efficient on a particular data set, but works poorly on another one. Therefore, the best way to treat a classification problem is to analyse the data, define the type of classification needed and its required overall performance and then to select adequate algorithms.
Practical applications in CRM needed today are mostly covered by mapping of data, e.g. texts, onto pre-defined classes, i.e. the supervised approach. If the amount of texts or e-mails per day is less than 300 and the amount of classes is low (<10), simple explicit pre-selected rules sets will work satisfactorily. In order to allow a more complex class structure with a bigger space of possible input texts, statistical or neural algorithms should be used. However, mostly Score and Naïve-Bayesian algorithms are sufficient.
During experiments with different text sources we discovered some pitfalls which can result in loss of time and money. One is the "garbage in - garbage out" pitfall. Here training data supplied by some business process, for example based on customer based web forms, plays an important role. It might be wrongly classified or just not distinguishable by the selected type of classification algorithm, or the training samples may not be statistically representative. We experienced the later case which is illustrated in figure 9. Imagine supplying the classification engine with the blue dots, representing the training data for the red curve. They neither represent the blue curve entirely nor contain much information about the red representative which the classification algorithm is supposed to learn. Even if you add more training data in the yellow shaded areas, the red curve will never be approximated, but the green one will. Thus, our text classification engine learned a model based on keywords with low significance, the "stop words" representing the "noise", leading to small differences in distance

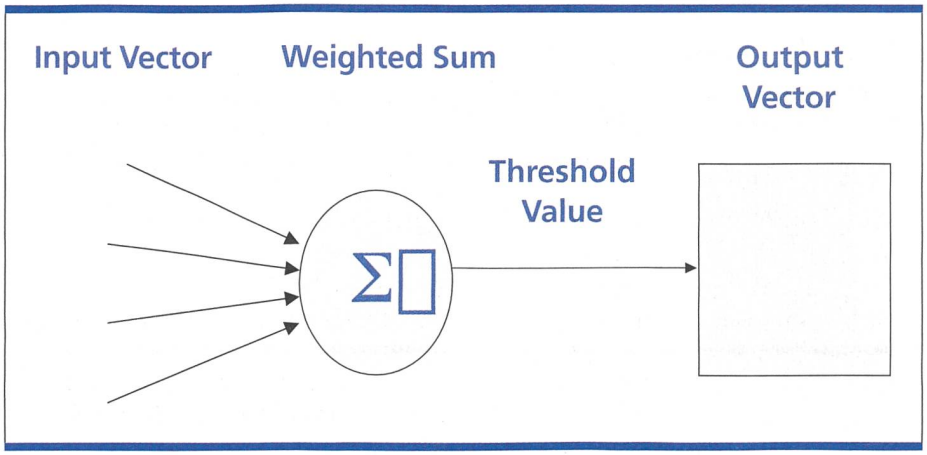

Fig. 8. Prototype of a simple neural network.

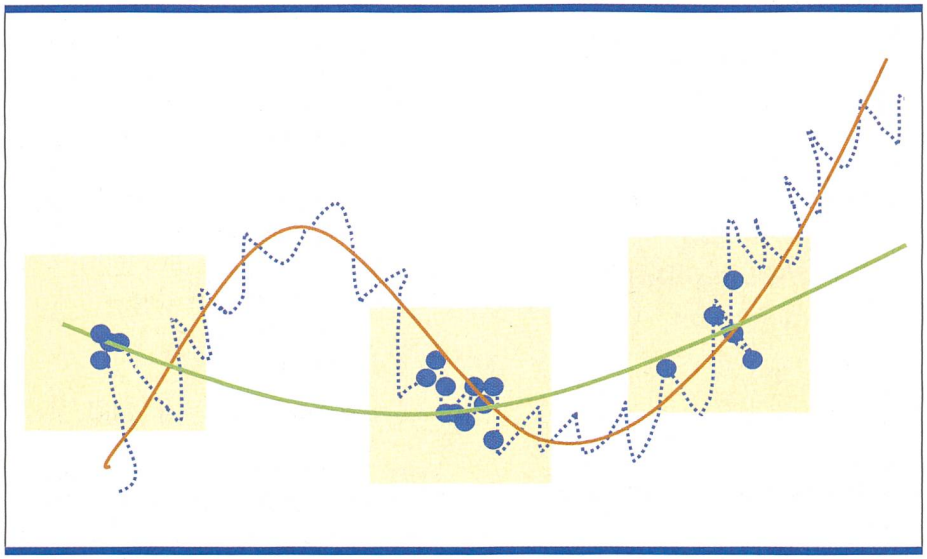

Fig. 9. Example for bad representative training data (bold blue dots on blue curve). Training leads to the wrong model (green curve), independent of the amount of data in the yellow shaded area.

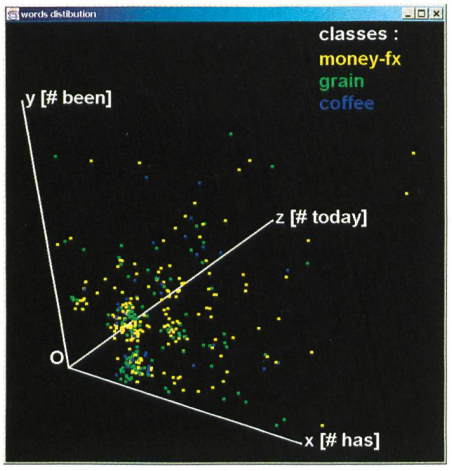

Fig. 10. Common case of not linearly separable key word counts (x, y, z axis) for three classes (money-fx, grain, coffee). The complicated correlation between the key word counts results in a complex and inefficient model.

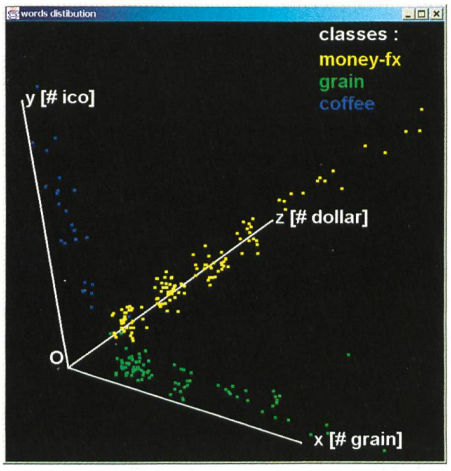

Fig. 11. Rare case for linearly separable key word counts (x, y, z axis) for three classes (money-fx, grain, coffee). As the key word counts are not correlated a simple rule-based decision tree can be applied.

between classes and therefore to bad performance.

The "garbage in – garbage out" pitfall effect can best be avoided by statistical analysis of the training data supporting the feature extraction process. Applying a feature extraction method developed in the context of the Exploration Programme we were able for example to reduce the benchmark input text vector size by a factor of 23, keeping the performance of the classification algorithm almost equal to 88% correct classification, on average. Subsequent tests revealed an excellent Generalisation Performance of the reduced model. In order to improve such a good model to reach 95% correct classification on average, different, more sophisticated methods as for example dictionaries have to be applied. But for present CRM applications this is usually not necessary. A better method would be to route texts with poor classification probability to a human agent mailbox where they are classified by hand, in order to produce new training data for model correction.

Another way to improve the performance of classification algorithms and reduce the time and effort in training is the identification of so-called "orthogonal features". They are linearly, and therefore easily, separable. Then a simple decision tree based on extracted features or their combinations could also show acceptable performance. 3D plots in figures 10 and 11 illustrate this orthogonal separable quality for key words. While in figure 10 no easy separation is possible, figure 11 shows the prototype of ideal features. The latter can be easily separated by a plane, resulting in a simple and efficient rule-based or score-based algorithm. Thus the right amount of significant keywords reduces the complexity of the classification engine and increases the Generalisation Performance. But often linearly separable features as outlined in figure 11 are not easily detectable without statistical pre-processing (see above). Without proper pre-processing the model in the classification engine becomes large and inefficient, often not better in terms of Generalisation Performance and hardly executable on standard computers. Accepting a higher misclassification rate could keep the model simple and processing speed and memory consumption low. Then pre-processing with

periodically updating the model could make standard rule based approaches, or if necessary simple Naïve Bayesian Networks in conjunction with existing platforms, perform better than the most cunning eCRM solution available on the market.

Conclusions

Summarising the lessons learned so far, the following rules hold for any classification engine:

- To reach acceptable classification performance, often statistical analysis and pre-processing are necessary. To do this successfully statistical expertise is crucial.
- The output of a classification algorithm is probabilities based on assumptions, not an almighty oracle or a human brain. Humans use a much more complex metric than machines and have general knowledge about the language itself, for example syntax, semantic and meaning.
- Using non-representative training data will lead to bad classification results.
- Commercial tools are not easily applicable to customer needs having already existing CRM facilities. Therefore the development of algorithms tailored to customer applications might perform better and at a considerably lower price.

Outlook

We intend to use our acquired practical knowledge to upgrade existing Swisscom eCRM facilities for example Call Centres or Web Services with the ability to automatically classify, route and auto-respond to customer e-mails or letters. The methods explored will also give input to current and future projects in web mining, fraud and intrusion detection. [10]

Text categorisation:

Y. Yang and X. Liu, A re-examination of text categorisation methods, http://www.cs.cmu.edu/~yiming

Classification techniques:

Editors: D. Michie, D.J. Spiegelhalter, C.C. Taylor, Machine Learning, Neutral and Statistical Classification, http://www.amsta.leeds.ac.uk/~charles/statlog/

Abbreviations

APM	Average Probability of Misclassification
CRM	Customer Relationship Management
eCRM	electronic CRM
GPRS	General Packet-Switched Radio Service
SVM	Support Vector Machines

Zusammenfassung

Bei automatischer Textklassifikation geht es darum, Texte aller möglichen Arten automatisch entsprechend ihrem Inhalt vordefinierten Klassen oder Kategorien zuzuordnen. Mit Textklassifikationssystemen kann ein weiterer Schritt in Richtung Automatisierung der Ordering- und Fulfilment-Prozesse erreicht werden. Aber auch im Bereich des Dokumenten- und Wissensmanagements leistet die automatische Textklassifikation wertvolle Unterstützung. In diesem Artikel wird ein Überblick über die heute verwendeten Methoden und ihre Anwendungen gegeben.

Stefan Burschka *is a physicist with special formation in quantum optics and microwave technology, computer science and economics. After positions in research and development he has been with Swisscom AG, Corporate Technology since 1998. His expertise is located in the area of CTI, text-classification, E-markets and intrusion detection with SW Agents. He is the head of the Autonomous Agents Lab and participates in SHUFFLE, an European IST-Project for UMTS Bandwidth brokering.*

Marcel Reitmann *studied theoretical Physics at the University of Berne and graduated in Quantum Field Theory in 1984. In 1986 he joined the Swiss PTT R&D department as a research engineer. During his long telecommunication career he has been working mainly in the fields of Network Performance and Quality of Service. Since 1997 he has worked in the areas of Customer Care and Customer Relationship Management as a research project leader.*

Sacha Varone *holds an Engineer degree in Mathematics and received a doctoral degree in applied mathematics from the EPFL (Ecole Polytechnique Fédérale of Lausanne) in 2000. He then joined Swisscom AG, Corporate Technology where he is involved in technology trends, data mining activities, text classification and business models.*

Frédéric Dreier *is an undergraduate student of computer science. His expertise is situated in the area of OO-Designs of complex SW, Text Classification and Data Clustering using methods of Artificial Intelligence and Artificial Life. Since1999 Frederic Dreier, besides his studies, has worked at Swisscom AG, Corporate Technology and has finished his obligatory industrial internship where he successfully completed several projects in the area of eCRM and Text Classification.*

SIEMENS

www.siemens.ch/jobs

Mehr als 150 Jahre Innovation haben uns zu dem gemacht, was wir heute sind. Das weltweit führende Unternehmen im Hightech-Bereich und eine der erfolgreichsten E-Companies der Welt. Hinter unserer globalen Präsenz stehen mehr als 440 000 Mitarbeiter in über 190 Ländern. Im gesamten Weltmarkt bewähren sich unsere in der Schweiz entwickelten Hightech-Produkte. Unser Innovationsgeist ist ungebrochen. Und wird es auch in Zukunft bleiben. Denn in den Entwicklungszentren von Siemens Schweiz AG arbeiten über 500 Ingenieure an den Produkten von morgen. Wie sehen Ihre Zukunftspläne aus? Sprechen Sie mit uns darüber.
Siemens Schweiz AG, Tel. 01-495 31 11.