Auto-Classification of Documents

Whitepaper by
Alexander Goerke, CEO Skilja GmbH

Document Understanding

For a long time document understanding has only been a research topic in computer sciences. Concepts and approaches to use computers and machine learning for understanding documents have been discussed on conferences and quite often this topic appeared in proceedings on text analytics or document analysis.

But in recent times many practical applications have become available that provide the basic functionality of understanding documents. Very often these applications are now used by enterprises to manage large amounts of incoming documents (especially paper), to allow rapid digitization and automatic recognition and distribution of documents. These solutions have been proven successful and a wealth of new concepts is now emerging providing much larger benefits to companies and end users making use of them.

The first goal of document understanding is to identify the function and the meaning of a document and its parts. Typically a document is written for a specific purpose which defines its function. An invoice is designed and created to notify a buyer on the goods bought and how much money needs to paid for them, along with some other information for accounting and tax purposes. All content of the invoice follows this function. Or an application form in a bank is used to collect all information that is needed to open an account. This document is normally very structured. On the other side an e-Mail (which is also a document) conveys information, opens a discussion and calls for action in a very unstructured way.

So the first step in document understanding is to identify the function and separate the documents to be processed accordingly. Typically this step is called “classification”. However this is only the primary classification or document type identification as more categorization according to a taxonomy can occur which do not have the purpose to define the function. It is therefore very important to distinguish these two types of classification as a lot of misunderstanding results from confusion between these. The function of a document determines the possible content and the information entities that can be found on it.

The second step where classification is used is the “page classification” that allows the system to identify the structure of a document and find relevant parts of it. For example a form might come
attached to a letter followed by a copy of an ID-card. These can be identified and interpreted separately. In the end page classification leads to a better navigation within a document and eventually document separation. Think of automatically adding bookmarks within a large PDF file.

The third step of understanding is to identify all information entities or predicates of a document related to the function. Typically only a few entities are needed and required (e.g. the amounts on an invoice) but preferably “understanding” means that all entities have been identified. To identify an entity (like a tax amount) it needs to be detected and a meaning must be attached. A number without a meaning is not a predicate. Only if all entities can be labeled with a meaning the computer system really understands the meaning of the document.

In the last step real understanding can take place if the entities are brought into the context of the document function, the purpose of the communication and the other entities. Typically a (business) document triggers an action. Context and correlation between the discovered predicates needs to be analyzed to determine which action. An e-Mail may contain a request to send some information back (which would be an entity “request for information”). But only in context with the rest of the e-Mail, the e-Mail thread or attachments it becomes clear which actions to perform.

As mentioned it is important to know the function of a document or any text to be able to understand its content. And with the knowledge of function (=purpose) and some content the system can take action and already has some kind of document understanding. This is what current solutions provide.

The first step in achieving document understanding is always classification.

Typical Use Case: Classification of incoming mail in an insurance company
Generic and specific classification

In principle it is possible to learn enough representative samples to create a classification scheme that is totally generic for a specific purpose. This is what humans do all the time. Reading a text and correctly classifying it manually into a given category is a task that is achieved by any data entry operator after a little bit of training. A machine learning algorithm can achieve the same thing faster and more precisely if only enough text samples are provided. This view is supported by recent research results from cognitive scientists who have found out that the acquisition of speech by babies is very similar to a Bayesian classifier. In fact babies learn to distinguish between different entities very quickly and are able to abstract from a concrete example to the generic feature of a concept (=class). It is proven that humans do not compare objects they see with stored images of objects but that they have a very effective feature reduction capability to abstract from the sample.

But it is also known that these systems fall short when the domain is very specific and the documents are very complex. It seems that in these cases manual data entry operators are much better and faster as they have additional knowledge about the domain and the world as such. The simple brute force statistical method fails more often with rising complexity.

Rule based classification

In the case of a low number of categories and high document complexity a carefully crafted rules based system is superior to any machine learning system. Even statistical systems are now moving towards introducing semantic rules and world knowledge (so called ontologies) to overcome these difficulties.
But rule based systems need to be created by system designers who are familiar with the subject matter and are able to code their knowledge into rules that rebuild this knowledge in a software application. Rule based systems have been very popular since a long time. The advantage of rule based systems is that they are totally controllable; the disadvantage is that their complexity rises faster than the complexity of the task.

The success of rules-based systems is dependent on the complexity of the document content and the size of the taxonomy as well as on a well coded collection of synonyms and keywords associated to each category. The challenge lies in creating and tweaking the rules for each category and keyword combination — which can be a lot of work for large taxonomies.

Generally rule based systems have become too complex and too expensive to maintain. Still they have their value. Skilja Classifier allows the user to combine rules with automatic classification in a hybrid system where the rules are manually defined and the dictionary is created automatically by training the system. This is only feasible for managing exceptions, exclusions and specific - well - rules. It is obvious that for any larger taxonomy with more than a few dozen classes any rule based system becomes unmanageable.

**Statistical classification**

The state of the art technology to be used is statistical and uses machine learning to identify a set of features in the text that are characteristic for a document category. Statistical classification models are mostly trained by supervised learning. During this process a set of representative samples is presented to the system. These samples are analyzed and relevant features are extracted automatically. Relevant means relevant for the type of object (in the case of text documents these are words, fragments of words and correlations between words, in the case of images these are image properties and forms) and relevant for a specific category – features that are present in one category but missing or less frequent in all other categories. A typical text page has about 250 words, of which maybe 20-30 have real significance and are not just grammatical filling words. To find the right and good features a significant amount of different characteristic text samples must be available and
prepared. Normally these are provided in the project setup as a learn set and then enhanced and changed using online feedback results from operation during production.

Features in the most simple case are just the words of the documents (this is called the “bag of words” approach). More sophisticated systems make use of stemming and lemmatization to normalize the words, so that for example “buying” and “bought” are recognized as the same feature related to a financial transaction. Others use trigrams or n-grams which reduce the words to a series of characters that are similar, and correlation between these elements. Finding general but also effective internal representations of natural language documents is the most important part of the process and decisive for the quality of the results.

Once the features are known a classification algorithm is applied to assign a weight to each feature. The weight determines the overall importance of a feature and the relevance for a specific category. For example grammatical stop words like “and”, “the” or “to” are so frequent in each document that they have zero overall importance and are eliminated completely. More meaningful words like “credit”, “mortgage” or “deed” might be present in the samples of several classes but only significant for one. The classifier therefore assigns weights to each of the features to mark the ones

The 10 most prominent features of the „Coffee“ news class

Features in the most simple case are just the words of the documents (this is called the “bag of words” approach). More sophisticated systems make use of stemming and lemmatization to normalize the words, so that for example “buying” and “bought” are recognized as the same feature related to a financial transaction. Others use trigrams or n-grams which reduce the words to a series of characters that are similar, and correlation between these elements. Finding general but also effective internal representations of natural language documents is the most important part of the process and decisive for the quality of the results.

Once the features are known a classification algorithm is applied to assign a weight to each feature. The weight determines the overall importance of a feature and the relevance for a specific category. For example grammatical stop words like “and”, “the” or “to” are so frequent in each document that they have zero overall importance and are eliminated completely. More meaningful words like “credit”, “mortgage” or “deed” might be present in the samples of several classes but only significant for one. The classifier therefore assigns weights to each of the features to mark the ones

<table>
<thead>
<tr>
<th>Detected Features</th>
<th>Feature String</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>COFFEE</td>
</tr>
<tr>
<td>1</td>
<td>International</td>
</tr>
<tr>
<td>2</td>
<td>COLOMBIA</td>
</tr>
<tr>
<td>3</td>
<td>quotes</td>
</tr>
<tr>
<td>4</td>
<td>export quotes</td>
</tr>
<tr>
<td>5</td>
<td>ICO</td>
</tr>
<tr>
<td>6</td>
<td>BOGOTA March</td>
</tr>
<tr>
<td>7</td>
<td>Coffee Organization</td>
</tr>
<tr>
<td>8</td>
<td>mlb bags</td>
</tr>
<tr>
<td>9</td>
<td>Organization ICO</td>
</tr>
<tr>
<td>10</td>
<td>coffee export</td>
</tr>
</tbody>
</table>

The 10 most prominent features of the „Coffee“ news class

Features in the most simple case are just the words of the documents (this is called the “bag of words” approach). More sophisticated systems make use of stemming and lemmatization to normalize the words, so that for example “buying” and “bought” are recognized as the same feature related to a financial transaction. Others use trigrams or n-grams which reduce the words to a series of characters that are similar, and correlation between these elements. Finding general but also effective internal representations of natural language documents is the most important part of the process and decisive for the quality of the results.

Once the features are known a classification algorithm is applied to assign a weight to each feature. The weight determines the overall importance of a feature and the relevance for a specific category. For example grammatical stop words like “and”, “the” or “to” are so frequent in each document that they have zero overall importance and are eliminated completely. More meaningful words like “credit”, “mortgage” or “deed” might be present in the samples of several classes but only significant for one. The classifier therefore assigns weights to each of the features to mark the ones

Word Cloud representations of feature hints for news class „livestock“
that are really important. It should be clear from this description that a significant number of typical samples are required in the learn set of each class to achieve a sufficient result.

Another useful way to look at the feature hints (these are not actually the features as they are complex mathematical expressions but a good approximation to what is used as features) is the word cloud that shows all features in alphabetical order with their importance corresponding to their font size.

When the relevant features are established they can be classified.

Classifiers that are used normally fall under one of the following families:

- Nearest Neighbor
- Support Vector Machines
- Bayesian Networks
- Maximum Entropy

The model that is created by the classifier contains all the knowledge about features, correlations and weights and stores it for runtime. At runtime the model is applied to unknown documents for classification. Now the features of the unknown document are extracted and compared to the stored weighted features. The result of classification is a list of confidences giving the correlation between the document and each class as shown in the figure above for the coffee document. The confidence is defined between 0 (no similarity) and 100% (very similar to class).

The confidence should not be normalized to the sum of all confidences so the sum of all confidences adds up to 100 but should give a true measurement of the correlation between the document and a class. A certain document might be adherent to two different classes with high confidence therefore it would be wrong to normalize the confidences in such way.

Often the training is iterative and the model is retrained with additional samples either offline or in production to get improved results.

**Image classification**

Image classification is an additional technology that is used to identify document pages by their layout. The good thing is, that only one single sample page is needed for training to be able to identify all similar layouts. Also image classification is very fast in the range of milliseconds as no OCR is needed. For image classification all kind of structures on a page are used, like logos, captions, lines etc. The precision of image classification is extremely high with almost no errors as a match is very clearly established. Of course image classification accounts for shift, skew and variation in resolution which is especially common for mobile images.

Image classification is used for identifying all known layouts prior to content based classification. When a layout is known its purpose can be determined. Image classification is used for identifying
all kind of forms in a document stream, but also invoices of a certain vendor (with constant logo and footer) or returned letters of your own company.

Card classification

A special variant of image classification is the identification of copies of cards, for example ID-cards, credit cards or driver licenses. Very often these copies are part of an application or account opening in an insurance or banking environment. In this case the location of the pattern of the copied card on the page can be anywhere and must first be identified. After finding one or more objects, the images are cut and classified with image classifier. As a result a rectified image of all card copies on a page can be presented to the user and to further processing.
Measuring Classification Quality

For an active production system, but also when the classification scheme is set up, it is very important to frequently measure the quality of classification. The goal is to create as few as possible errors in classification (also called false positives) as these can severely impact business processes. It must be clear that automatic classification systems, in the same way as human operators, will always make a certain amount of errors. By measuring the performance the error can be quantified and minimized. The error rate from automatic classification must at least match human error rate which has been measured to be about 4% for complex classification tasks.

The question is, how classification quality can be measured. Often vendors give ONE number: “Our success rate is 95%!“ This is a useless statement. First it does not say how the success is measured. Secondly it does not say anything about errors or false positives. Quality of any recognition system always needs to be characterized with at least two numbers.: Quality of classification is measured using precision and recall testing a reference set that has been manually defined and tagged. While precision gives the percentage of documents that have been classified correctly with respect to all classified documents, recall is the number of documents classified into a class with respect to the total number of documents in the reference set of this class.

In a simple graph, these sets can be shown: The quality is first determined for one class. All documents of the reference set in the light green ellipse are divided into two sets: The documents that belong to class A – and the documents that do not belong to class A. These sets are defined by the red line and are established by manually tagging the documents in the reference set. The classifier, however, will perform a slightly different separation depicted by the blue line. After the classification of all documents in the set we arrive with 3 subsets: “a” contains all correctly classified documents, “b” the incorrectly classified documents and “c” the documents that should have been classified into class A but were not recalled. Using these sets the following quality measures are defined:

\[
\text{Precision} = \frac{a}{a+b} \\
\text{Recall} = \frac{a}{a+c}.
\]
To obtain a global value of quality for the complete taxonomy the weighted averages of all classes are used (also called micro average).

In document input, a precision of more than 95% is desirable in most cases. Normally this cannot be achieved without introducing a threshold that limits the results to documents that are above the predefined confidence for classification. The threshold can be defined for each class and has the effect to create another subset of documents, the rejected or unknown documents which are shown in blue in the graph.

The threshold will reduce the false positives (set b). The higher the required confidence, the fewer documents are erroneously classified. Unfortunately the threshold will also reduce the number of correctly (but less confidently) classified documents as is shown in the graph. A good classifier is characterized by reducing the false positives faster than the true positives thus increasing precision faster than decreasing recall.

In a pragmatic setup the designer of the classification scheme will set precision to the value as obtained in manual classification which is about 96%. The resulting recall (which could be around 80%) is a direct measure for the productivity gain of the automated classification system as all the rejects must be manually classified as before. In certain cases it is necessary to have a higher precision (e.g. for sensitive documents) or a higher recall might be preferred.

The threshold allows you to choose the desired scenario. The relation between precision and recall for the complete document set can be shown in the precision-recall graph. Here the respective percentages of false positives and correctly classified documents in a tagged reference set are depicted as a function of the selected threshold. With a threshold of zero recall and precision are obviously identical as every document that is not recalled correctly in one class will appear as a false positive in another class. With the introduction of thresholds false positives will be avoided but also the recall will be reduced. A good classifier shows a steady gradient of the curves with no bumps and will allow the system designer to precisely select the threshold as required by the business needs of the organization.
10 rules for creating a successful mailroom classification project

Automatic, context based classification for mailrooms has proven to generate significant ROI and acceleration of processes in the last few years. But also failures and disappointments have been seen.

We have managed and monitored many of these projects in the past and would like to share 10 golden rules derived from our experience to make a mailroom classification process successful:

1. **Plan enough time to prepare:** Changing the way how business processes originate is a severe organizational change for the company. Although work in the mailroom is often considered of minor importance, this is the area where everything starts. Any error here has significant effects on quality of service and response time. Any improvement will ensure that responses to requests are faster and customer satisfaction is maintained. So don’t begin with implementation right away. Forget about technology for a moment, put aside tools and spend time to analyze existing processes upfront. Take them and define clearly which of them you can automate and what would be the best way to approach this task. Document the findings and have them signed off by the customer.

2. **Involve stakeholders:** Classification drives and initiates business processes. In each organization you find existing stakeholders and subject matter experts (SMEs) who are familiar with the processes and have done manual sorting for years. They know the documents that arrive, they know explicit rules but also a lot of implicit procedures. Identify the SMEs and invite them to the
team that defines the classification scheme. You can learn a lot from them. Create incentives for their active participation so you get access to the hidden knowledge which they need to share with you to make you understand their business. Often a simple, straightforward valuation of the work they have been doing until now is enough to get them involved.

3. **Define goals:** Clearly define goals and have them signed off by the team. Set the expectation and explain to the team what classification can achieve – and what it can’t! Very often clients have unrealistic expectations on the performance of the new system and wrong assumptions about the manual process at the same time. Depending on the kind of documents manual classification creates as much as 5% of errors and misclassification. It makes sense to hold a general educational session about classification technologies and the preconditions for successful classification. If clients are introduced (on a high level) to the technological foundation, they will understand better that even an automation rate of 70% with an error rate of 3% can be a big success. And that 99% are not realistic. Make sure that everybody understands that quality can only be measured statistically and that it makes no sense to focus on single documents that might have been misclassified.

4. **Use good data sets:** Get a large and representative set of documents from several weeks to account for changes in content by weekday. Typically 1,000 to 2,000 documents need to be reviewed. Work with the SME team to sort the documents manually and make sure to understand why they sort them as they do. Often the reasons they give for sorting contain valuable hints on exceptions that need to be coded as rules and cannot be trained automatically. Create 3 data sets from these documents: A training or development set, a test set and a reference set and tag them with the correct classes. The last one will be touched only for finally measuring the quality to achieve sign off. The test set will be continuously used to measure effects of training and rules during development.

5. **Use clean data:** Make sure that you clean the documents before using them for classification. Documents contain “noise” like footers, general terms and disclaimers that are not relevant for the content. These need to be removed by prior analysis (e.g. font size) as they will for sure mislead the classifier. For e-mails make sure to remove text from old threads (which can be identified by indents) that often cascades over many responses.

6. **Start small:** Do not attempt to solve the complete categorization problem at once. Focus on the main categories and start with them. In the beginning use the 10 major classes (classes with the highest number of documents) and create a working scheme. Then expand by adding more classes in steps of 10 or 20 while continuously measuring quality.

7. **Use hierarchy:** If the classification software allows to use hierarchies (it should!) make intelligent use of it. Main classes can be easily identified and subsequent classification can further break them down into the desired target classes corresponding to business processes. As the differences between documents become smaller and smaller the closer the topics are, it is easier to handle these distinctions in a hierarchy. Together with rule 6 hierarchy provides and easy and straightforward way to reduce complexity and build the scheme step by step.
8. **Run continuous tests:** Make sure that you test after each change. Quality can be assured if you know what you are doing. Running automated tests after each change allows you to better understand the reasons for possible deterioration. Searching these reasons later might prove very difficult.

9. **Ramp up production:** Don’t attempt to process the complete volume from day one when you start production. Rather start with a fraction of the volume (10%) or, if possible, run automated classification in parallel for some days. This allows the business users to accommodate themselves with the system and allows you to correct errors that shine up in production only. Ramp up the volume step by step until you process the complete volume after a few weeks.

10. **Monitor in production:** Measure and monitor quality in production. To achieve this you need statistics that show how many documents had to be classified manually and for how many the class was changed by users. Deterioration starts with day one. This is not necessarily the fault of the classification software but more often due to slow but steady changes in the document content over time. By monitoring the system it can be tuned during production to stay up to date. If you are using a system with automated learning, monitoring is essential to find out if the quality really goes up.