# Toward Cloud-based Classification and Annotation Support

Tobias Swoboda[1], Michael Kaufmann[2] and Matthias L. Hemmje[1]
[1]*Faculty for Mathematics and Computer Science, University of Hagen, Universitätsstraße 47, 58084 Hagen, Germany*
[2]*Lucerne University of Applied Sciences and Arts, Technikumstrasse 21, 6048 Horw, Switzerland*

Keywords: Text Categorization, Evaluation, Architecture, Cloud Cost Analysis, Internet of Services, XaaS, Cloud Services.

Abstract: Manually annotating content-based categories to existing documents is a time-consuming task for human domain experts. In order to ease this effort, automated text categorization is used. This paper evaluates the state of the art in cloud-based text categorization and proposes an architecture for flexible cloud-based classification and annotation support, leveraging the advantages provided by cloud-based architectures.

## 1 INTRODUCTION AND MOTIVATION

The European project *Computer-Aided Process Planning for Sustainable Manufacturing Environments* (CAPP4SMEs) aims to enhance the competitiveness of European companies by providing cloud-based integrated process planning environments along the supply chain. *Process-oriented, Knowledge-based Innovation Management* (German: *Wissens-basiertes Prozess-orientiertes Innovatonsmanagement,* WPIM) satisfies this need by providing a cloud-application supporting process-oriented semantic knowledge representation. Within WPIM, processes are represented in a semantic fashion. WPIM also provides the means to annotate additional documentation to every process (Vogel, 2012) This additional documentation is usually stored in documents. Companies moving their processes into the WPIM environment usually already have documentation on their processes. Annotating them to the semantic process representations in WPIM is a considerable manual effort that can be eased by using *text categorization*. One goal of the CAPP4SMEs project is the creation of a cloud-based software environment. The comparison of existing cloud-based text categorization services are a logical step to provide text categorization within the WPIM context.

For this planned research, we have identified the following questions: How can *text categorization* be distributed in the cloud in a flexible way? How can this flexibility be used in order to generate highly effective and efficient *classifiers*?

This paper is structured as follows: Section two introduces the formal definition of text categorization, introduces four available approaches and compares them. Section three introduces our model architecture to implement a flexible cloud-based classifier. Section four focuses on the possible evaluation for this proposed classifier architecture, while section five explains the possible implementation of our approach. Section six draws conclusions and describes future prospects of this ongoing research.

## 2 STATE-OF-THE-ART AND RELATED WORK

The aim of *text categorization* is to ease the knowledge acquisition bottleneck. According to (Sebastiani, 2002), this bottleneck is defined as the lack of available domain expert time to assign categories to documents. When these domain experts assign documents to categories, they create a binary *target function* $\Phi'$: $D \times C \rightarrow \{true, false\}$ that indicates whether or not a given document, $d \in D$, belongs to a specific category, $c \in C$.

In text categorization, *classifiers* are constructed. Formally, classifiers are defined as function $\Phi$: $D \times C \rightarrow \{true, false\}$ that are constructed in a fashion that $\Phi'$ and $\Phi$ coincide as much as possible.

There are two basic approaches in constructing classifiers. The first approach is the construction of



Swoboda, T., Kaufmann, M. and Hemmje, M.
Toward Cloud-based Classification and Annotation Support.
In *Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016) - Volume 2*, pages 131-137
ISBN: 978-989-758-182-3

*rule-based classifiers,* which rely on disjunctive normal form (DNF) formulas to check on the occurrence of certain terms within the text. In text categorization, a *term* can be a simple word or a more complex regular expression describing entire sentences. The second approach in classifier construction is machine learning, which consists of computational methods that use past experience to make accurate predictions (Mohri et al., 2012).

Machine-learning-based classifiers have the decisive advantage over rule-based classifiers in that they can learn from examples. The construction of effective rule-based classifiers requires experts on the construction of such classifiers, as well as domain experts. On the other hand, machine-learning-based classifiers require a set of pre-classified documents as examples from which to learn. These documents are commonly referred to as initial corpus Ω. This initial corpus is usually divided in a training set *Tr* and an evaluation set *Es*. The first is used to actually train the machine-learning-based classifier, while the latter is used to examine its effectiveness. While efficiency describes how quickly and resource efficient a machine learning algorithms categorizes documents, *effectiveness* is a measure of quality on how much Φ' and Φ coincide.

According to (Mell and Grance, 2011), cloud computing is defined by a set of essential characteristics. These are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service.

(Mell and Grance, 2011) also represent cloud computing as a layer model. The top layer, Software as a Service (SaaS), is essentially software that can be accessed by end users or other software services through web portals or APIs.

The intermediate layer, Platform as a Service (PaaS), is a system that provides software developers with an environment to implement and deploy their applications. It consists of a programming-language-level environment with a set of well-defined APIs.

The bottom layer, Infrastructure as a Service, (IaaS), provides computational resources in the form of virtual machines or data storage space.

(Weinhardt et al., 2009) compared cloud computing with grid computing and stated that cloud computing is systematically coupled with a business model, thus making a pricing model a prerequisite for a service to be considered as cloud computing.

Based on (Weinhardt et al., 2009) and (Creeger, 2009), the usage of cloud computing in all layers has the following advantages:

- Flexibility. Cloud computing services are already there and can be utilized when needed. Less capacity planning is required.
- Shift from capital expenditure (CapEx) to operational expenditure (OpEx) in a way that the services are paid for as they are used. There is no need for high up-front investments to buy and set up the necessary server environment. If the service is only utilized a little, then costs are saved.
- Easy integration of different services over internet APIs.
- Enforceable service level agreements (SLAs) with cloud providers reduce operational risks.

The following description of the state of the art in cloud-based text categorization assesses three SaaS and one PaaS offerings.

## 2.1 SaaS and PaaS Classifiers

Accessible at https://www.meaningcloud.com, this classifier provides a combination of statistical document classification with rule-based filtering. Statistical analysis is used to define categories based on example documents. One can also create manual fixed rules for fine-tuning. As categories are defined by providing example documents, this SaaS can be regarded as a hybrid machine learning/rule-based classifier. The utilized algorithms and system architectures are black-boxed and, therefore, unknown to the cloud user. The API is accessed by HTTPS POST requests. A user can either upload the text that needs to be categorized in the HTTPS POST packet (in that case being limited to 8192 characters) or provide a URL from where meaningcloud can load the text. As HTTPS is used, this communication is SSL protected. Texts are never transmitted unencrypted and are not stored within the meaningcloud service. Replies are either XML- or JSON-encoded lists of categories for the provided text. Meaningcloud contains a set of pre-trained out-of-the-box classifiers for certain sets of topics. One can also use an own initial corpus Ω to create a custom classifier with user-defined categories. When enrolling in the meaningcloud service, a user is provided with an API user key. This key must be stated in every HTTPS POST request. (Meaningcloud documentation, 2015) A user then consummates a subscription available in differently sized packages. Subscription packages limit categorizations per month and categorizations per second. Standard packages range from free with 40,000 categorizations per month and 2 per second

to 42,000,000 categorizations per month and 15 per second for $999.00 per month. For more intense usage, custom contracts can be arranged. (Meaningcloud pricing, 2015)

The Bitext SaaS is accessible at https://www.bitext.com. It is a rule-based classifier employing complex terms. The API is also accessible by HTTPS POST requests. The only way to transmit the text for analysis into bitext is to transfer it with the HTTPS POST request. Therefore, texts are limited to 8,192 characters. Responses are XML, JSON or CSV encoded. The out-of-the-box bitext classifier comes with a set of categories that is product marketing related and available in English, Spanish, Portuguese, Italian, French, German, Dutch and Catalan. By the usage of HTTPS, all communication with bitext is SSL encrypted. Custom classifiers can be generated by bitext in a project with the user's organization. In such a project, the user pays a project-specific price and needs to collaborate with bitext personnel to create the custom classifier. (Bitext professional service description 2015) When accessing bitext, a username and corresponding password must be stated with every HTTPS POST request. A cloud user then consummates packages. One package costs $995.00 and allows for 1,000,000 categorizations. It has to be consumed within 6 months. A 300-day free trial is available. (Bitext API documentation 2015)

The textwise SaaS is available at http://www.textwise.com. It employs the *Semantic Gist* similarity search engine. In this system, example documents define their categories as a similarity comparison which is performed with every document that needs to be categorized. It can be regarded as machine-learning classifier. The service is accessed by HTTP POST requests. In every request, the user must state a usage authentication token. One must also provide the URL to the source from where the text needs to be extracted. Responses are encoded in XML, JSON, RDF or TagClouds. Because regular HTTP is used, the communication with the textwise API is not encrypted. Textwise is only available in English and consists of one classifier that categorizes to the set of categories defined by the open directory project classification scheme, a taxonomy consisting of 2,047 categories. There is no public pricing information on textwise. (Textwise API documentation 2015)

Other than the previously mentioned SaaS classifiers, the Google Prediction API is a PaaS classifier. It can be used as API for an application that is developed in the *Google developers console* PaaS environment. The Google prediction API can be called from within the *Google developers console* or external systems via HTTPS. Authentication is handled by OAUTH 2.0. To implement the OAUTH authentication, an application within the PaaS environment is required. Texts and the initial corpus Ω must be stored in the Google cloud storage service.The Google Prediction API provides a completely trainable and flexible machine-learning environment. Besides text categorization, it can also be used to create numeric predictions based on training data. The employed algorithm is black-boxed and is, therefore, unknown to the user. The first 10,000 predictions per month are free. Additional predictions are paid in 1,000 prediction packages for $0.50 each with additional costs for the cloud storage and Google developer console PaaS environment. (Google prediction API documentation 2015)

## 2.2 Comparison of the Mentioned Approaches

This chapter evaluates the usefulness of the introduced cloud-based text categorization technologies from a *WPIM* perspective. The following criteria are utilized:

- Customizability of the classifier to specific sets of categories.
- Accessibility from the existing *WPIM* cloud-based application.
- Language support.
- Text size limitation.
- Security.
- Pricing.

Every aspect is quantified in a simple grading system with poor grades (1), medium grades (2) and good grades (3). The following table summarizes the results. Details on every aspect can be found in this chapter's subchapters.

Table 1: Comparison of existing cloud-based text categorization services.

| Aspect | Meaning-cloud | Bitext | Text-wise | Google Prediction API |
|---|---|---|---|---|
| Customizability | 3 | 2 | 1 | 3 |
| Accessibility | 3 | 3 | 3 | 2 |
| Language support | 3 | 2 | 1 | 3 |
| Text size limitation | 3 | 1 | 3 | 3 |
| Security | 3 | 3 | 1 | 3 |
| Minimum categoryzation price | $.000024 | $.000995 | n.a. | $.0005 |

An effectiveness evaluation is not a part of this paper, due to the lack of classifier customizability.

Such an evaluation of the sufficiently customizable services is a crucial future work on this paper, as classifier effectiveness is the most important factor in usefulness. Disregarding effectiveness, meaningcloud provides the most useful service for the lowest price per categorization and is, therefore, a suitable candidate for further testing. Meaningcloud and the Google Prediction API can be trained with any custom set of categories. Bitext customization requires the time of bitext personnel and domain experts. Both are sparse and are, therefore, costly. This knowledge acquisition bottleneck creates an obstacle on the customizability of the service. Textwise cannot be customized at all.

Meaningcloud, bitext and textwise are accessed via HTTP(S) POST requests. Every request requires the user to provide credentials in form of a key, a token or a combination of username and password. The Google Prediction API requires an application to run in the Google platform to handle authentication. As the existing WPIM does not run in the Google platform, the necessity of this application lowers the accessibility grade. Due to their flexible trainability, the Google Prediction API and meaningcloud theoretically provide all required languages. Bitext is more limited, but still suitable in a European context, as most western European languages are supported. Bitext, however, does not support any Slavic or Scandinavian languages. Only supporting the English language textwise is the most limited system in this regard. Bitext has a limit of 8,192 characters per text. This is a rather severe limitation as, for example, this paper has more than thrice that size and, therefore, cannot be categorized with bitext. All other services have no practical limits in text size. The Google Prediction API uses HTTPS and OAUTH2.0 as security mechanisms. Meaningcloud and bitext use SSL-encrypted HTTPS. Textwise only uses HTTP and transmits the authentication token as clear text.

Regarding pricing, the prices of productive categorizations are compared. Free usage quotas are not taken into consideration. Meaningcloud provides 42,000,000 monthly categorizations for $999.00, which puts the minimum cost per categorization at $.000024. A bitext package allows 1,000,000 categorizations over a timeframe of 6 months for $995.00, creating a minimum cost per categorization of 0.000995. Google charges $.50 for 1,000 categorizations, putting the cost per categorization at $.0005.

A common gap among these services is a lack of an integrated effectiveness evaluation capability providing details on the individual categories. Half of the accessed services also lack the ability to create custom categories without provider interaction, which violates the NIST cloud characteristic of on-demand self-service. Our approach attempts to mitigate these shortcomings by simultaneously leveraging already-existing systems in a *classifier committee*.

# 3 MODEL

Based on (Sebastiani, 2002), a *classifier committee* follows the idea that $k$ experts may be better in passing judgment than one. Therefore, such a committee is a system in which $k$ different classifiers $\Phi_1,\ldots,\Phi_k$ are tasked with categorizing the same document and then combining their results properly. The easiest combination is a simple majority vote. Classifier committees work best if the individual classifiers work as independently as possible. A cloud-based architecture can easily implement this by only developing the central committee and by recognizing only text categorization SaaS applications as its members. As most services evaluated in this paper require the text to be transferred to the cloud application, this central system requires the possibility to grant access to the *WPIM* document store.

(Swoboda, 2014) examined the approach to utilize continuous expert feedback to every machine-learning-based classifier output and retrain the classifier accordingly. To do so, a human domain expert would examine a categorized document and either approve of the automated categorization or correct it.

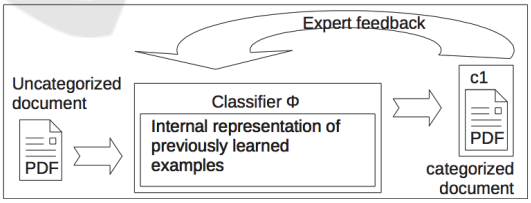


Figure 1: Expert feedback process in (Swoboda 2014).

This way, the initial corpus Ω will grow in size, as documents that are not humanly categorized are not considered part of Ω. The enlarged Ω is then used to perform a retraining of the classifier by *n-fold-cross-validation*, a process in which the initial corpus Ω is divided into $n$ subsets of about equal size. One of these subsets is reserved as *Es* while the classifier is trained with the remaining $n$-1 subsets. *Effectiveness* values are obtained by validating the classifier using the reserved *Es*. After validation, the

effectiveness values are stored and the process is repeated, reserving the next subset for evaluation purposes. After *n* iterations, the *training set* that produced the most effective classifier is chosen as the productive training set for the classifier.

It is possible that n-fold-cross-validation and internal committees are already part of the examined text categorization services. As they are black-boxed, the user is oblivious to this.

Our approach is the combination of continuous feedback with n-fold-cross-validation and a classifier committee consisting of a configurable number of cloud-based classifiers. After initial training with a sufficiently sized Ω, the system provides domain experts with automatically categorized documents for approval or correction. After Ω has sufficiently grown in size, the system triggers a new n-fold-cross-validation process.

In order to implement this, only a core system with access to the central document store implementing the n-fold-cross-validation and feedback processing system is necessary. This can then, in turn, call the APIs of available cloud-based text categorization services.

But why stop at only using existing cloud-based classifiers? (Sebastiani, 2002) organized text categorization in 3 phases upon which (Swoboda 2014) built an additional phase 0 and implemented a flexible system allowing for different solutions and algorithms in every phase. The phases are:

- Phase 0: Text extraction. The transformation of different document types into plain text.
- Phase 1: Feature extraction. The transformation of texts into feature vectors that are comprehendible by most machine learning algorithms.
- Phase 2: Feature selection. Reduction of the feature vector's size to speed up and enhance the effectiveness of machine-learning algorithms.
- Phase 3: Machine-learning-based text categorization.

An additional phase between Phases 2 und 3 is possible to enhance the effectiveness of the overall classifier. We propose the following model for a cloud-based text categorization system: The central *Cloud Classifier Committee* System *C*3 consists out of a set of *macro classifiers* $\Phi^M$, a *Corpus Manager CM* and an *Expert Feedback Handling Module EF*.

$$C3 = \{\{\Phi^M{}_1, \ldots, \Phi^M{}_n\}, CM, EF\} \quad (1)$$

Every *macro classifier,* $\Phi^M$, consists of a committee, *CO,* and a specific $\Omega_t$, with which it has been trained.

$$\Phi^M = \{CO, \Omega_t\} \quad (2)$$

*CO* consists of a set of *classifiers* and a specific combination function. $f_c$.

$$CO = \{\{\Phi_1, \ldots, \Phi_n\}, f_c\} \quad (3)$$

Every Φ can either be a cloud-based text categorization service or a combination of different Phase 1 to Phase 3 cloud services.

As described in the feedback process, every $\Phi^M$ can generate categorizations. One $\Phi^M$ is selected as the productive *macro classifier* based on its effectiveness results. Its categorizations are used in the *WPIM* system and then queried to experts by *EF*. After sufficient expert queries, $|\Omega|$ is increased enough to start an *n-fold-cross-validation* process. This process creates a new $\Phi^M$ as $\Omega_t$ is updated. Only the $\Phi^M$ with best effectiveness scores is used as a productive classifier. This means that if the new and updated $\Omega_t$ results in a classifier with poorer effectiveness, an older $\Phi^M$ is used.

# 4 POSSIBLE EVALUATION

An automated effectiveness evaluation is an integral part of *C*3, as the *n-fold-cross-validation* process needs to decide which combination of *Tr* and *Es* yielded the best results. The same evaluation function can be used compare one $\Phi^M$ with another $\Phi^M$. It can also be used to evaluate the *effectiveness* of every cloud-based classifier that is used.

There are different commonly used *effectiveness* measures (Sebastiani, 2002), focusing on different aspects of *TC*. Usually a confusion matrix is generated per category *c* counting document assignments.

Table 2: Confusion matrix per category *c*.

| | | Output Φ | |
|---|---|---|---|
| | | in *c* | not in *c* |
| Definition Φ' | in *c* | TP (True Positive) | FN (False Negative) |
| | not in *c* | FP (False Positive) | TN (True Negative) |

As there are usually more than two categories, these results can then be either *microaveraged* or *macroaveraged* to generate global effectiveness measures. In *microaveraging*, confusion matrix results are added before effectiveness scores are calculated once for the entire classifier. In *macroaveraging*, the effectiveness values are calculated for every category before calculating a global mean of these results. Both methods give

different results for the same classifier. Choosing the correct one for *C*3 depends on the distribution of Ω as, for example, the ability of a classifier to work well on categories with few training samples is emphasized by *macroaveraging* and much less by *microaveraging* (Sebastiani, 2002) Common measures are precision, π (4), and recall, ρ (5).

$$\pi = TP / (TP + FP) \quad (4)$$

$$\rho = TP / (TP + FN) \quad (5)$$

The first is a measure of how precise categorizations to certain categories are while the latter is a measure of how many documents that should have been assigned to a category actually were assigned to it. It is advisable to use a combination of both as a measure to decide classifier effectiveness.

# 5 IMPLEMENTATION

The model is based on the idea of combining the available classifiers or subservices that can potentially form a classifier into committees. There are two classes of users for *C*3: Domain experts and administrators. Domain experts provide their knowledge by manually assigning documents to categories. Administrators integrate *C*3 with the available classifiers and subservices. To do so, the required API calls must be configured to *C*3 along with the necessary authentication mechanisms to access these services. Additionally, administrators provide crucial configuration values: Firstly, a measure of how much |Ω| needs to increase before the next n-fold-cross-validation is triggered; secondly the actual *n* in the cross-validation process and, thirdly, a definition of which combination of *micro*- and *macroaveraged precision* and *recall* is used to determine classifier effectiveness. All these values have a potential impact on the cost of such categorizations. All the examined freely trainable classifier cloud services don't charge for training. The $n*(n\text{-}1/n)*|\Omega| = (n\text{-}1)*|\Omega|$ evaluations, however, take up the available quota and increase the time needed to retrain. All categorizations and triggered retrainings in C3 will be logged in order to enable billing and usage reporting.

Each of the previously mentioned phases can be implemented differently. The combination of these different implementations creates a unique classifier. For the implementation of every phase, available SaaS APIs can be utilized from a central control point. If these prove to be insufficient for the overall purpose, PaaS or IaaS offerings can be utilized to implement these phases individually. The central control point can be implemented in a PaaS or IaaS environment with access to the existing WPIM document store by a HTTPS-based interface. Phases 0 to 2 create binary or real-value vector representations of the document. As most machine learning algorithms require such feature vectors (Mohri et al., 2012), they are stored in a vector store that's part of the proposed core classifier manager solution. Using this feature vector store, the first phases only have to be executed once, speeding up the n-fold cross-validation process. This abstraction of text to feature vectors provides additional benefits: A layer of security and enhanced speed for the overall system as potentially sensible and lengthy documents are only transferred to phase 3 cloud services as a vector representation of their content instead of humanly readable text.

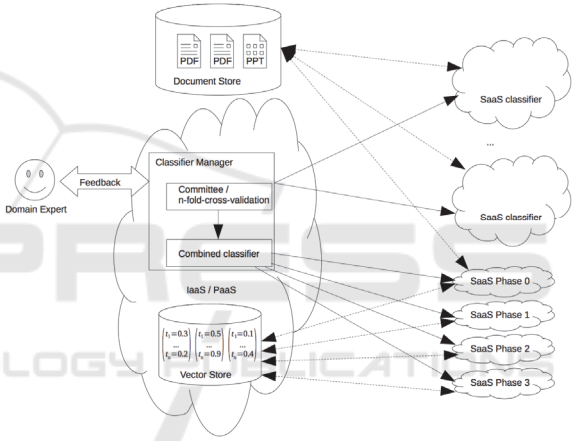


Figure 2: Proposed cloud-based classification architecture.

SaaS classifiers that implement phases 1 to 3 can use the central document store HTTPS interface to poll the texts that need to be categorized from this central document store. This central document store has the advantage that documents are stored in a single location, thus avoiding consistency issues. A disadvantage in comparison to storing the text with the SaaS classifier solution is that the same text has to be transmitted once with every n-fold cross-validation iteration after sufficient feedback was given. As it is part of WPIM, the central document store is, strictly speaking, already a part of a SaaS system. Because the utilized cloud services are executed asynchronously, the core system needs to generate cross-validation schedules that write instructions for the cloud services into queues to be executed and subsequently combine the results.

The *C*3 implementation will be accessible through a HTTP(S)-based API in order to make it

capable of being integrated in other cloud-based systems. It will also possess a web-based GUI that accesses the API in the background. There is no inhibition to implement a self-service onboarding mechanism that would allow a new customer to become a tenant with his or her own *C*3.

# 6 DISCUSSION AND FUTURE PROSPECTS

This paper compared four cloud-based text categorization services. The comparison focused on non-functional aspects of the different cloud services. Disregarding effectiveness, meaningcloud and Google prediction API offer the best services for low pricing. Based on gaps identified by a comparison of four cloud-based text categorization services, we propose an architecture combining cloud-based text categorization in a classifier committee with different feature extraction and machine-learning algorithms while utilizing continuous domain expert feedback.

This proposed architecture has the advantage that it automatically evaluates its own and its' components effectiveness. Automatically choosing the best solution, it can potentially provide a highly effective categorization solution. Another advantage is its flexibility to combine different cloud services and their intrinsic non-functional advantages. The system will intrinsically provide a platform to evaluate the impact of different approaches during the multiple phases of text categorization on effectiveness. The system is therefore easy to use for domain experts. Because *C*3 uses a set of existing classifiers, the costs to use *C*3 will be more than the sum of the utilized services. If the *C*3 cloud service provider has multiple customers, the cost-efficient big usage quotas of the underlying cloud providers can be used to provide a more affordable categorization service.

A possible disadvantage lies in its distributed nature across the Internet. Response times and sub-service availability cannot always be guaranteed. Another possible disadvantage is the transmission of potentially sensible documents to different systems.

Future prospects in this field are an effectiveness evaluation of already available SaaS classifiers. Besides this evaluation, an implementation of the proposed C3 architecture and effectiveness evaluation of this system is a logical next step. In order to leverage the multi-phase implementation, cloud-based text extraction and machine-learning services will be compared and their impact on effectiveness evaluated.

# REFERENCES


CAPP-4-SMEs description of work, 2012

Vogel, T. Wissensbasiertes und Prozessorientiertes Innovationsmanagement WPIM, doctoral thesis, Hagen, Germany, 2012

Sebastiani, F., Machine Learning in Automated Text Categorization, ACM Computing Surveys vol. 34 (2002), 1-47

Mohri, M. Rostamizadeh, A., Talwalkar, A., Foundation of Machine learning, MIT Press, Cambridge, Massachusetts, USA, 2012

Mell, P., Grance, T., The NIST Definition of Cloud Computing, National Institute of Standards and Technology, Gaithersburg, USA, 2011

Weinhardt, C., Anandasivam, A., Blau, B., Stößer, J., Business Models in the Service World, IEEE Computer Society, Issue No.02 – March/April (2009 vol.11) 28-33

Creeger, M. CTO Roundtable - Cloud Computing - The age of cloud computing has begun. How can companies take advantage of the new opportunities it provides? Communications of the ACM: CACM; BD 52.2009, 8, pp. 50-57, New York, USA, 2009

Meaningcloud documentation. Accessible online at https://www.meaningcloud.com/developer/text-classification/doc/1.1/what-is-text-classification (accessed June 27, 2015)

Meaningcloud pricing. Accessible online at https://www.meaningcloud.com/products/pricing/ (accessed June 27, 2015)

Bitext API documentation. Accessible online at https://www.bitext.com/wp-content/uploads/2014/11/Bitext_API-Reference-Manual_EN.pdf (accessed June 27th, 2015)

Bitext professional service description. Accessible online at https://www.bitext.com/text-analysis-technology/text-analysis-cloud-services-api/customization/ (accessed June 27, 2015)

Textwise API documentation. Accessible online at http://www.textwise.com/api/documentation/api-services/category-service (accessed June 27, 2015)

Google Prediction API documentation. Accessible online at https://cloud.google.com/prediction/docs (accessed January 12, 2015)

Swoboda, T., Towards effectivity augmentation of automated scientific document categorization by continuous feedback, Master Thesis, University of Hagen, Germany, 2014