Text Classification based Filters for a Domain-Specific Search Engine

Sebastian Schmidt[1,2] Multimedia Communications Lab, Technische Universität Darmstadt, Germany [3]

Steffen Schnitzer[2] Multimedia Communications Lab, Technische Universität Darmstadt, Germany

Christoph Rensing Multimedia Communications Lab, Technische Universität Darmstadt, Germany

[1] Corresponding authors

[2] Equally contributing first authors

[3] E-mail: sschmidt@kom.tu-darmstadt.de, Tel: +49-6151-166115, Address: Multimedia Communications Lab, Rundeturmstraße 10, 64283 Darmstadt, Germany

Abstract

Domain-specific search engines exist in various fields, providing additional value by exploiting knowledge of their respective domains. One common mechanism used are filters which allow narrowing down the search results based on pre-defined filter categories. In this article we exploit the usage of a text classification system for the creation of these filters. The approach is tailored to work in large-scale settings with reduced amounts of manually annotated training data and hence enables a cost-efficient roll-out of new filters. An initial annotation study resulted in a corpus which was used for an off-line evaluation of the approach giving insights into the effect of the system's parameters. Finally, a large online evaluation was executed together with a provider of a domain-specific search engine. This article presents important aspects that need to be taken into consideration when implementing text classification-based filters in the industrial setting of a domain-specific search engine.

Keywords: Search engines, text classification, annotation study, active learning

1 Introduction

The amount of digital information is growing at an enormous pace. In August 2014, around 179 million [1] active Web sites were reachable where the majority does not consist of single pages but hundreds or thousands of sub-pages. While multimedia content such as video and audio is gaining increasing importance as a relevant type of content, text still seems to be the prevalent media type.

After the rise of the Internet, search engines have evolved in order to provide an efficient access to available information. While the market is dominated by generic search engines, in particular Google[4], an increasing number of domain-specific search engines have come up. Common domains for these domain-specific search engines are products, restaurants, hotels, job offers or scientific publications. In comparison to generic search engines, domain-specific search engines bear advantages by exploiting knowledge of their respective domains in order to improve the search experience for the user. Both recall[5] and precision[6] of the search can be improved compared to generic search engines:

- Recall can be improved by incorporating search results from all or a high fraction of relevant Web pages (maintaining an index of relevant Web sites in the background)
- Precision can be improved by giving the user the possibility to search not only based on string values over complete documents but to filter the search results according to their needs in the particular domain

In this article we focus on the latter. Filters are a pre-defined number of attributes that can be combined with each other in order to narrow down the result set of a search. These filters can be generic or domain-specific. Examples for generic filters are file-format, top-level-domain or language.

[4] http://www.google.com

[5] *Recall* describes in the context of search engines the percentage of relevant Web pages that were found by a search query out of the total number of relevant pages

[6] *Precision* describes in the context of search engines the percentage of relevant Web pages out of the total number of Web pages found

In contrast to these generic filters there are domain-specific filters such as the price for a product, the geographical region a job offer is relevant to, the color of a car or the number of rooms a hotel has. These filters need to be explicitly integrated into the search interface and the underlying filtering logic cannot easily be ported to another domain.

Some of the filters can best be implemented by rule-based approaches, e.g. prices can be found by searching for a number followed by a currency symbol. This holds for all information with a small or restricted number of distinct expressions which are independent of the context (e.g. if there are several prices on one site, the price of interest can only be identified based on the textual context). In contrast to this, some information cannot be explicitly found in the Web pages themselves or the language allows for a huge variety of expressions.

In this article, we focus on filters that require more advanced techniques for realization since the textual part of the document through which the filter value can be determined is of rather complex nature. The present work exploits text classification using machine learning techniques as a means to build these filters. The deployment of the used approach takes place in the industrial setting of a job search engine. The goal is to use the textual content of the job offers (a.k.a. documents) as input for text classification and use the available class labels as filter criteria for the search engine. Therefore, the approach has to decide for any job offer if a particular filter (a.k.a. class label) should be evaluated positive or negative. A filter matches for a job offer if and only if the text classifier decides to assign the filter class to the job offer.

As time to market of newly identified filter is critical, it is required that the filter can be deployed fast. In addition to this, domain-specific search engines often start as start-ups or small-sized companies and do not have the manpower and/or financial resources to manually annotate large amounts of text documents as required by classical supervised machine learning classification approaches. At the same time, a certain classification quality needs to be met [2] for achieving user satisfaction. Based on this observation, we examine an approach that relies on active learning, which allows for a fast and cost-efficient deployment through reduced manual effort i.e. the time spent on annotations. This approach has been presented before [3] but is now, for the first time, evaluated during runtime in an industrial setting instead of using historical data in a lab environment for evaluation. To enable a meaningful deployment of the approach, we introduce a new parameter for the integration in the industrial setting.

The actual deployment of the approach in an industrial setting enables a detailed description of the processes involved and gives insights on problems that occur in such a real-world scenario. Through the online evaluation, the active-learning setting is not only tested for its theoretical performance but for actual applicability for different problems within the daily routine of an industrial partner. The presented experience, gained during the experiments, allows a repeatable application of this text-classification approach for document filtering.

The remainder of this article is structured as follows: after this introduction, an overview on related work is given. Afterwards, in Section 3 the application scenario on which this article focuses is presented; this presentation is required in order to understand the specific needs in this scenario. The following section presents the approach to and the results of an annotation study executed within the application scenario. Section 5 describes the concept of the adapted approach while Section 6 describes the implementation within the application scenario. The results of the evaluation using the data from the annotation study are presented in Section 7. Finally, the work is concluded in Section 8.

2 Related Work

The present work combines the application area of domain-specific search engines with the technique of active learning. To the best of our knowledge, there is no existing work which has examined this combination so far. In this section some aspects of these two fields are highlighted. Furthermore, we highlight the aspect of balancing of training data, since this is relevant for our work as well.

2.1 Domain-specific Search Engines

This section presents some approaches to the implementation of domain-specific search engines. Hanbury et al. [4] define a domain-specific search engine as “a search engine that specifies one or more of the following five dimensions: 1. subject areas, 2. modality, 3. users, 4. tasks, 5. Tools, techniques and algorithms required to complete the tasks”. Within the scope of this work, we concentrate on the specification of the subject area since the web sites to be retrieved all origin from one subject are.

A consistent usage of Semantic Web[7] technologies from content providers resulting in machine-readable data would simplify the creation of domain-specific search engines [5]. Since the Semantic Web has not prevailed until today, search engine providers cannot rely on semantic annotations of Web documents but need to understand the content automatically by other means in order to make it retrievable in real-time.

Some approaches to domain-specific search engines have been presented where the user’s query is enriched with domain-specific keywords, forwarded to a generic search engine and the returned result is presented to the user [6] [7]. This enrichment with keywords in order to receive only relevant results can be seen as a coarse-grained filtering, but in general the approach is hardly feasible since it entirely relies on the quality of the generic search engine and, in addition to this, most generic search engines would nowadays block these queries coming from a single source at high volume. Further, many domains and filters are too complex to be modeled by keywords only. A domain-specific search engine that does not rely on a generic search engine and builds generic rules based on given keywords was presented by Kruger et al. [8]. Another way towards an optimal user experience is the ranking of the result list, which can be enhanced by adding domain-specific features [9].

2.2 Active Learning

Active Learning has been a popular research topic for about 20 years now. Motivated by the fact that obtaining labels is expensive, the general idea is to provide only labeled instances with a high information value as training instances to a supervised classifier, resulting in fewer instances to be labeled manually. The instances to be labeled are selected by the classifier in its current state, assuming that the knowledge about these labels improves the future accuracy of the classifier. A

[7] One of the aims of the Semantic Web vision by Tim Berners-Lee [16] is the machine-readability of the web. This requires a consistent annotation of Web pages with semantic labels on word-level yielding a closer interlinking of stored information.

variety of approaches for the selection of these instances has been proposed; an overview on these was given by Fu et al. [10].

In general, active learning can be and has been combined with various classification techniques, e.g. Bayesian models [11] and Support Vector Machines [12], which have been identified as being most suitable for text classification tasks [13].

Also, different combinations of active learning and ensemble learning, which exploits a set (ensemble) of classifiers, have been proposed. Examples are the combination of various classifiers from two different classification approaches where the ratio between the classifier types in the ensemble is adapted during run time [14], the combination of different classifiers of the same type but trained with different feature selections [15] and the combination of different classifiers of the same type but trained with different subsets of the data [3].

2.3 Balancing

It has been shown that the quality of a classifier trained on unbalanced data sets can be improved by balancing the training corpora [25]. A more balanced data set yields a more robust classifier when used for training. Common methods are random oversampling and random undersampling where the former involves a duplication of the minority class while the latter involves discarding of instances of the majority class. Running evaluations with data sets with different severities of imbalance it has been shown that these two approaches yield best results among various different balancing approaches for Support Vector Machines [18]. The approach presented in this article is based on Support Vector Machines which indicates the high potential of these two techniques.

3 Application Scenario

In order to gain insights into the feasibility of the proposed approach as a filtering means we have implemented it at a domain-specific search engine. Below, we explain the characteristics of this scenario.

Kimeta[8], founded in 2005, is one of the market leaders in the German online job search market. The Web site has about 20 million page views through about 2.5 million visits per month. Every day, 80,000 to 100,000 new job offer documents enter the system via a Web crawling process and need to be processed so that users can retrieve them as results of their search. This processing does not need to happen in real-time but the time span from first publication of the document until delivery to the end user is a key performance feature. The main sources for the documents are (i) company Web sites, (ii) job offer markets and (iii) newspaper Web pages. The documents from these three sources display different characteristics. Each company presents the documents in their own (corporate design) layout with varying structure reaching from pure flowing text to well-structured tables. In contrast to this, job offer market pages provide a standardized layout. Finally, newspaper pages can be characterized by rather short job offers since the offers are often published parallel in print and the price for a print publication depends strongly on its length. This variety shows that the scenario imposes high requirements on the document processors in order to index all different kinds of documents reliably. The text classification system used must be highly robust in order to allow for classification of various kinds of documents.

[8] http://www.kimeta.de/

Kimeta provides several different filter groups on its Web site, ranging from hours of work (full time, part time) or mode of employment (regularly employed, internship, temporary job,…) to the functional area of work (Consulting, Controlling, IT, Medicine, R&D….). Analyses of data on another job search platform have shown that such a filtering possibility is of high relevance for job searchers [19]. In this work we focus on binary filters which either match or do not match a document. Depending on the concrete filter, the fraction of job offers which is supposed to be matched versus the fraction of job offers which should not be matched differs clearly. Classification approaches cope best with a balanced distribution and are challenged by such an unbalanced setting.

In order to provide these filters for the domain-specific search in the domain of job offers, Kimeta relies on several different approaches to identify the characteristics of the job offers, such as keyword-based filtering and text classification. The creation of such filters is expensive due to the need of domain experts which analyze the filter scenario and create either the keyword-based filters or initiate the creation of a text classification based filter. To build robust text classification-based filters, a sufficient amount of documents has to be annotated. Because annotations of a single annotator can be biased, it has become a de-facto standard in research to carry out the annotation not only by a single annotator but by at least two annotators [24]. The result of combining these single annotations is considered to be more representative for the common understanding of a class label. Furthermore, this allows for an early identification of ambiguities in the definition of classes and yields finally a "clean" gold standard dataset by incorporating only instances with an agreement on their annotations. Kimeta's pre-studies have shown that non-trained annotators have difficulty deciding on the correct filter decision. Therefore, one big challenge in the application scenario is how to perform a process which enables a profound and repeatable annotation by mainly non-experts to keep the financial burden low. The design considerations, settings and the results of an annotation study are presented in the following section.

4 Annotation Study

This section describes the annotation study, which was performed in order to provide evaluation data for our approach, to understand the complexity of the classification problem and to develop a process to cope with that complexity during the annotation phase.

4.1 Setup

In the beginning, three different binary filters were selected for this study. This decision was made based on the company's previous experience with these three filters where it has been discovered that a keyword-based implementation does not yield satisfying precision and recall. Furthermore, the filters were selected to describe highly varying types of concepts. The first filter describes a job offer's field of activity while the second filters for a characteristic which is independent from the field of activity and the third filter describes a mode of employment. The filters selected are the following ones:

1. Service/Customer Support (*S*)
2. Research & Development (*R&D*)
3. Full time job (*FT*)

Throughout the annotation, a group of three people was involved in the process, a domain expert and two non-experts. After the selection of these filters, the domain expert wrote a short informal definition for each of the filters. This definition was afterwards discussed within the group which led to some refinement of the definitions. In order to check for completeness, each of the group members annotated a set of 50 documents for each of the three filters based on the previously determined definition. For each of the documents and each of the filters, the annotators had to choose between three different labels: “Positive”, “Negative”, “I do not know”. During the annotation process, the annotators searched for cases where they incorporate implicit knowledge and opinions not given in the filter definition. For example, after a first definition for the FT filter was mutually agreed on, some further characteristics had to be agreed on e.g. whether jobs as a freelancer, jobs done in home office or working-on-site should be considered to be *FT.* The annotators were asked to write down such cases of implicit knowledge in order to have it included in the final annotation guidelines. All documents with inconsistent annotations were discussed leading to a further improvement of the filters’ descriptions and recorded in the annotation guidelines. In previous work [20], such an iterative improvement of annotation guidelines has been shown to be beneficial. The resulting guidelines consist of the following elements: (i) a short and informal definition of the filter, (ii) some indicators and examples for positive instances, (iii) some indicators and examples for negative instances, (iv) borderline examples together with their correct annotation and (v) a hand-written decision tree for the respective filter. The total amount of man-hours for the setup of one filter including the iterative creation of the definition, the selection of sample documents, discussions and documentation was 5:30h for the expert and 3:40h for the non-experts

For the remaining annotation, the annotators were told to adhere strictly to the created definitions. For the creation of the evaluation corpus, 300 random documents were selected. All of these documents were annotated by two independent annotators (the non-experts) for each of the three filters. For the documents where the two annotators did not agree on the annotation (one selecting “Positive” and the other “Negative”) or both annotators were uncertain (selecting “I do not know”), the domain expert was consulted as third annotator.

4.2 Results

The annotation of the 300 documents by a single annotator took around 2h. The complete creation of the annotated documents, including selection of the documents to annotate by the expert, annotation, resolving disagreements and file conversions took on average 2:40 man-hours for the expert and 4 man-hours for the non-experts.

The results of the annotation study are presented in Table 1. The rows represent the different filters while the columns represent the different levels of agreement. A perfect agreement was achieved when the two initial annotators agreed on the annotation. An agreement was achieved when one of the initial annotators decided for “Positive” or “Negative” while the other one decided for “I do not know” or the same. Also the cases where both initial annotators decided for “I do not know” but the third annotator decided for “Positive” or “Negative” were counted as agreement. The most prominent reason for selection of “I do not know” were titles of the document which did not totally match with the content and the annotators were not sure on whether they should rely on the title or on the content. A disagreement was assumed if either one voted for “Positive” and the other one for “Negative” or if both of them voted for “I do not know”. All these cases were resolved by the third annotator.

Cohen’s Kappa [17] is a robust statistical measure for the inter-annotator-agreement between two annotators since it subtracts out the by-chance agreement. The inter-annotator-agreements of the two initial annotators are as follows: $\kappa_{R\&D}$=0.745, κ_{FT}=0.659 and κ_{S}=0.520. One can conclude from the data that the decision for filter *R&D* was the easiest one for humans while the annotation for *S* led to some confusion. The "substantial agreement"[9] on the annotation, in particular for the classes *R&D* and *FT*, gives an impression on how the thorough and concerted filter definition with the domain expert helped the non-experts to cope with the given problems.

Table 1: Percent agreement for two annotators

Filter	Perfect Agreement	Agreement	Disagreement
R&D	92.67%	96.33%	3.67%
FT	94.33%	97.00%	3.00%
S	84.00%	88.00%	12.00%

As already mentioned in Section 3, the filters in the application scenario are highly unbalanced. Out of 300 documents, 36 were finally labeled as positive for the filter *R&D* (12%). For the filter *FT* 277 documents were labeled as positive (92.3%) and for the filter *S* 50 instances were labeled as positive (16.7%).

Besides the experiences with the presented iterative process of annotation through a mix of experts and non-experts in a real-world active-learning scenario for text-classification based filters, we introduce a new tuning parameter for the approach which will be explained in the following section.

5 CENFA Approach

Within this work we make use of CENFA [3]. Full details of this approach are given in the respective article, but its big picture is presented in Figure 1 which shows the combination of an ensemble classifier which is only trained initially ("the base classifier") and a single SVM which is iteratively re-trained, applying active learning ("the specialized classifier"). As shown [3], the re-training of the specialized classifier is significantly faster than a re-training of the complete ensemble while at the same time, the system provides better results compared to a system trained with a similar number of documents that were sampled randomly instead of selecting them by the use of active learning.

The classifier returns a confidence value for each classified document which is given by either the base or the specialized classifier and derived by the voting scheme of the ensemble or the single confidence value respectively. In this binary class setup, the confidence value for the more probable class per document is used which results in a range of [0.5;1.0] for the confidences. The setup presented by Schnitzer et al. [3] uses one single *confidence threshold* to determine whether the classifier marks a document for human annotation (steps 3/8), or provides a confident annotation decision (steps 7.1 - 7.3).

[9] A Kappa value between 0.61 and 0.80 is considered as "substantial agreement" according to Landis and Koch [26]

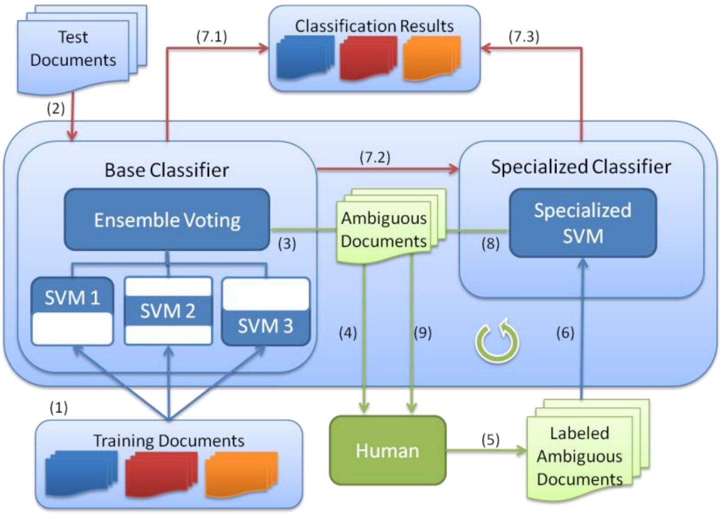

Figure 1: Classifier Setup as described in [3]

In contrast to the previous work, these two confidence thresholds are decoupled into the *annotation confidence threshold* (*act)* and the *decision confidence threshold* (*dct*), respectively. This adaptation was chosen to allow for the deployment in a large-scale setting of the industrial application in contrast to the lab environment of the previous work. The two parameters can now be set independently of each other. This change is motivated by the fact that the number of documents in the application scenario is very high and the *act* needs to be adapted to the actual throughput in a scenario while the *dct* needs to be adapted based on the characteristics of the data.

The documents to be annotated for the active learning step are detected based on the *act*. Its range of [0.5;1.0] is determined by the range of possible confidences of a binary classifier. If the overall classification confidence of the base classifier for a document is smaller than *act,* the document is stored for annotation by human in the first iteration. Such a document is assumed to be ambiguous and will be referred to like this throughout this article since the base classifier did not identify any agreement on its class label. These annotated documents serve as training data for the specialized classifier. That means that a low *act* yields fewer documents to be annotated and hence less training data for the specialized classifier and vice versa. Implicitly, the percentage of documents that need to be annotated can be tuned by the *act*.

To determine whether the decision of the complete system is carried out by the base classifier or the specialized classifier, the *dct is* used. In scenarios with high throughput this decoupling allows to “produce” only a reasonable amount of documents to be annotated while also sending documents with a lower confidence to the specialized classifier. This advantage can only be exploited if $dct \geq act$.

6 Realization in the Application Scenario

The approach described above holds several challenges for a realization with the industrial partner. The process on how such a system is initially deployed and maintained throughout the iterations as well as the integration of the workflow in the company is described in this section.

At first, the steps described in Section 4 are followed to create a well-defined filter’s description. To gather the training documents for the respective filter, a representative selection of job offers is taken from the current stream of documents and stored in a separate database. These documents are then annotated independently by the two non-experts involved in the filter definition and the results are stored back into the database. The results are then analyzed towards the agreement of the annotators, so that the domain expert can draw the disagreed-on documents from the database to impose a final decision as described before. Now that the basic training set of documents is given, the model for the particular filter is created and can be deployed in the ensemble classifier of the classification engine.

The following process is carried out for each iteration. Throughout the day, the stream of newly discovered and therefore yet to be classified documents runs through the classification engine and the high confidence results per document are added as a characteristic of the job offer which enables the filtering in a later search. During this phase, the documents for which the classification results stays below the defined threshold (*act*) are added to the separate database and marked for post-annotation in the respective filter. Now non-expert annotators can draw sets of documents for annotation from a certain filter, for the purpose of evaluation we ask the annotators to draw the 100 most ambiguous documents in each iteration. Due to the well-defined filter's description, annotators who were not involved in the initial creation of the filters require a very short training period and can therefore draw documents from various filters. The annotation of the 100 most ambiguous documents takes between 0:30h and 1:30h depending on the filter and the experience of the annotator. The system has to ensure that the two independent annotations required for a document are carried out by different individuals. Afterwards, the domain expert retrieves the small set of documents which require a final decision and judges them. The model is re-trained with the now completely annotated set of ambiguous documents, and deployed in the specialized classifier of the classification engine which takes less than ten minutes. With this step, one iteration is finished and the next one can start afresh. Hence, the iteration takes 24 hours for collecting a sufficient amount of ambiguous documents and around 2 hours for annotation, data transformation and deployment of the updated classifier. Since the system runs continuously, the time frame for the collection of ambiguous documents can in general be extended arbitrarily and annotation and re-training can happen whenever it is favored.

The inter-annotator-agreement of the first two iterations is presented in Table 2. One can observe the drop in terms of agreement for each of the classes compared to the values during the initial annotation phase and within the two iterations. This shows, that documents with a high ambiguity for the classifier (which are the ones passed to the annotators for post-annotation) are also hard to classify for the annotators.

Table 2: Cohen's Kappa for the inter-annotator-agreement of the two annotators during the two iterations

Iteration	$\kappa_{R\&D}$	κ_{S}	κ_{FT}
1st	0.55	0.48	0.40
2nd	0.38	0.24	0.40

7 Evaluation

This section describes the process of the system's evaluation. After introducing relevant measures and explaining the overall setup, the results of the tuning experiments in an offline setting are presented. Afterwards the results of integrating the approach into the live system are shown. It should be noted that within this article, no comparison to related approaches is presented. These comparisons to other active learning approaches were presented before [3] and it was shown that CENFA is addressing the trade-off between classification quality and reduced manual annotation effort as well as a performant classification and re-training. The focus of the evaluation in this article is rather the experience in the industrial setting with the particular approach.

7.1 Evaluation Measures and Setup

As introduced before, this work copes with an unbalanced application scenario, where the number of positive and negative instances is highly different. In such a setting, the usage of the weighted F-measure, as often done, represents a certain trade-off between precision and recall. However, the given industrial scenario describes an information retrieval problem, where the scenario-dependent metric of choice is the precision of the classifier, since false-positives are a major issue in search engine results and should be avoided. In other scenarios, depending on the concrete setting, the major aim might be to tune towards other metrics like a high specificity or a high sensitivity of the classification model. In order to get information about the tuning of both and allow for an unbiased evaluation in an unbalanced setting, the Receiver Operating Curve (ROC) is commonly used. The *Area-under-the-Curve (AUC)* is the respective singular numerical value representing the overall discriminative performance of the model [21] [22].

Therefore, we use as performance measurements the scenario- and balance-independent macro-averaged AUC of the ROC, as well as the scenario-dependent and industrially motivated macro-averaged precision.

Another relevant key aspect of the classifier is the number of ambiguous instances identified during the iterations since this corresponds directly to the number of instances that need to be manually annotated. Ideally, the system would identify only small amounts of ambiguous documents which lead to a huge improvement in the iteration step.

Throughout all evaluations the corpus consisting of 300 instances per class, gathered during the annotation study, is used as gold standard[10]. The feature set used consists of the 10,000 most used unigrams using the term frequency – inverse documents frequency (tf-idf) weighting scheme. The number of bagged SVMs was fixed to 10. This value is a trade-off between the accuracy of the single SVMs which need enough instances to be trained properly and a sufficient number of SVMs in order to gain a robust ensemble [23].

7.2 Tuning Experiments

Before the system is deployed in the industrial setting, two different tuning experiments are executed in order to find the best *act* and evaluate the influence of balancing for the examined application scenario. For these tuning experiments, a stratified 4-fold cross validation was applied for each of the three classes resulting in a split with 225 training instances and 75 test instances. Furthermore, for each of the given training set sizes, 10 random samples were drawn out of the 225

[10] The complete corpus and the corpus created during the post-annotation (see Section 6) can be found online: http://www.kom.tu-darmstadt.de/fileadmin/Externer_Bereich/Downloads/software/Paper_TextClassificationBasedFilters_EvalData.zip

available training instances. The given results for each of the sizes are obtained from the resulting 40 evaluation runs.

Effect of Balancing

As mentioned in Section 2.3, previous experiments have shown that a balanced training set yields a more robust classifier. Oversampling as well as undersampling has been identified as the most appropriate techniques for balancing. This section presents the results for a comparison of unbalanced vs. balanced training data. In order to have datasets with a comparable common size across the different classes we combine for the balanced training set oversampling with undersampling so that the total amount of available training instances is at 225. The test documents were left unmodified.

Figure 2 presents the average AUC together with its standard deviation for the base classifier (without any iteration) for the different filter classes comparing balanced unbalanced data. It can be observed that the overall trend is an increase in terms of AUC with increasing number of training instances. Comparing the balanced setting to the unbalanced setting, the balanced setting provides slightly better results for small amounts of training data than the unbalanced setting. In general it has to be noted, that the standard deviation is relatively high.

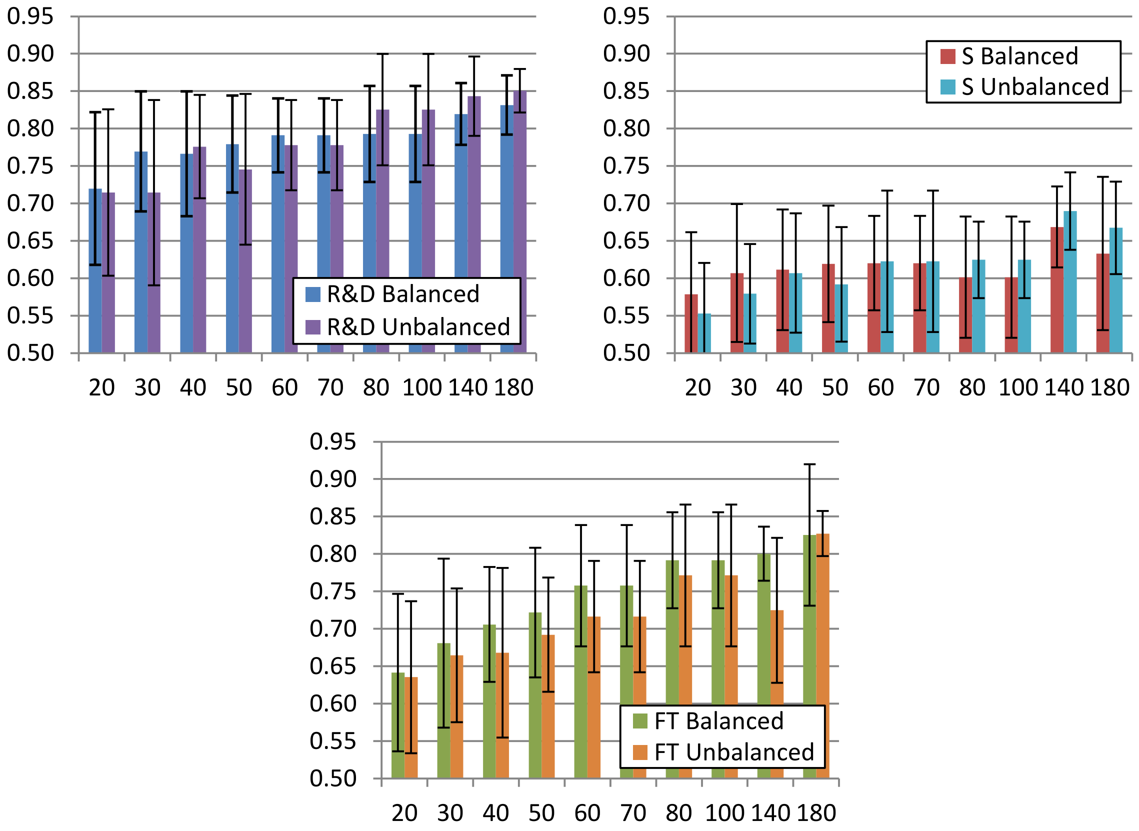

Figure 2: AUC and standard deviation for the Base Classifier with varying numbers of training instances for different classes

Impact of Annotation Confidence Threshold

A key factor for the deployment of the approach in a real-world setting is the amount of documents found to be ambiguous since these need to be annotated which requires manual effort, i.e.

employees' time. The variable *act* was introduced in order to allow for a tuning of the fraction of documents to be annotated manually. Figure 3 shows the impact of the *act* on the fraction of documents identified as ambiguous with a varying number of documents used for training the base classifier using the balanced dataset. The steady increase in all curves proves that the overall goal of the introduction of *act* was achieved. From the varying fraction between the different classes one can infer a different complexity of the classification into these classes. Interestingly, the class which was most complex for the ensemble to classify (class *S*) is also the class with the lowest inter-annotator agreement (see Section 4.2). Further, it can be seen that a larger number of initially annotated documents leads to an increase in the flattening of the curves. This shows the value of the initial effort for the training of the base classifier. In particular in settings with a high throughput, it pays off to train the base classifier properly so that it is more confident about the classification and fewer documents need to be post-annotated.

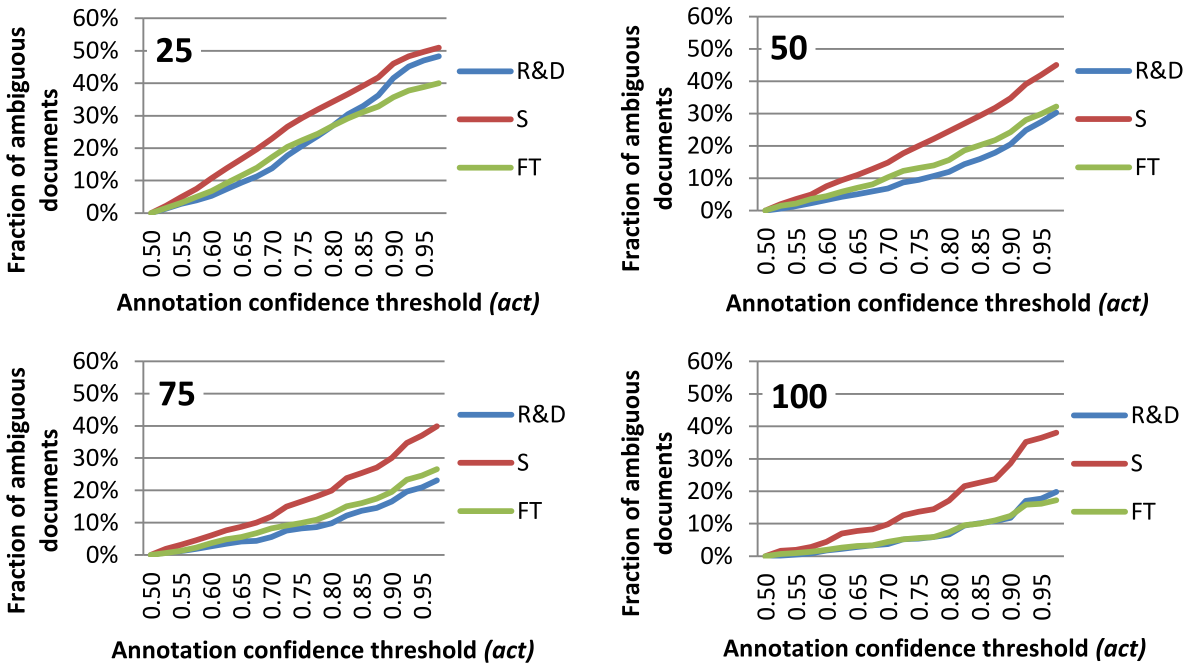

Figure 3: Fraction of documents to be annotated against the *act* parameter for the different classes. The numbers in the upper left corner denote the number of training instances for the base classifier

7.3 Experiments within the Industrial Setting

The evaluation in the application scenario requires to run the classifier for a certain time, collect documents that are ambiguous to the particular classifier, annotate these documents manually by two to three annotators and re-train the specialized classifier with the annotated documents. In order to examine the behavior over several iterations, these steps need to be repeated. Since the set of ambiguous documents depends on the set of initial training documents and the manual annotation of ambiguous documents is the most expensive process step (in terms of human resources), it is impracticable to run the same experiment with different folds of the training data. We therefore had to select one fold for each of the three classes (consisting of 225 annotated, balanced training documents), built the classification models out of these and relied on the results for these three initial training sets only instead of averaging over the different folds. The remaining 75 annotated documents are used as test data. A randomly chosen part of the documents of the daily throughput were classified by the models and depending on the *act* selected for annotation.

These post-annotated documents were then used to train the specialized classifier. Afterwards, the classifier consisting of base and specialized classifier was evaluated. This iterative cycle of classification, post-annotation, re-training and evaluation was repeated two times.

The results of the execution in the application scenario allow examining four different aspects of the classification task. To gain further insight about the impact of *dct* and *act*, we gathered different data through the iterations. We also analyze the number of documents which were selected for the annotation by the system and discuss their class distribution. The performance of the classifier is analyzed as well by describing the behavior of precision and AUC throughout the iterations.

Impact of Annotation Confidence Threshold

For the first iteration almost 30,000 documents were classified. The fraction of documents which were selected for annotation in the first iteration can be seen in Figure 4. Similar to the tuning experiments the higher the *act* is chosen, the more documents are selected for annotation. Because of the large throughput in this setting even the smallest possible *act* (0.5) led to more documents than planned to be annotated.

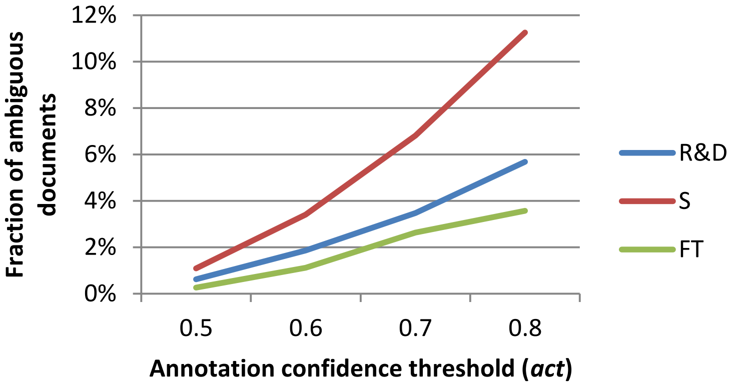

Figure 4: Fraction of ambiguous documents in iteration 1

For the second iteration a second set of documents was used, because some of the documents of the first set were now included in the training set for the model. Here the model did not select enough documents for annotation, and therefore the number of documents for classification was increased to slightly more than 80,000 documents. The same number of documents was used in a 3rd iteration which was stopped after having identified the number of ambiguous documents. The results for the iterations can be obtained from Table 3. The fraction of documents that are selected for annotation decreases from the first to the second iteration. The *act* was set to 0.8 for demonstration purposes only, using a smaller value would not demonstrate the decrease during iterations that clearly.

Table 3: Fraction of ambiguous documents across the iterations (*act* = 0.8)

Iteration	*R&D*	*S*	*FT*
1st	5.68%	11.25%	3.57%
2nd	0.25%	0.38%	0.09%
3rd	0.01%	0.02%	0.01%

Impact of Decision Confidence Threshold

The impact of the *dct* is shown in Figure 5, where the *act* was fixed to a value of 0.5 in order to keep the number of documents that need a post-annotation relatively low. That means that only such instances were chosen for annotation that appeared hardest to classify for the ensemble. For the different classes the performance changes differently. The figure shows the results for the different *dct* for the two iterations that were executed completely *(<class> it 1* and *<class> it 2*). Furthermore, the results for the base classifier only are presented *(<class> base*). Since in this case no instances are routed to the specialized classifier for decision, the evaluation metrics stay constant. Finally, the results of *the extended classifier*, as introduced before [3], are shown for the two iterations *(<class> ext it 1* and *<class> ext it 2*). The *extended classifier* an ensemble of SVM similar to the base classifier being trained with all available training instances after iteration 1 or iteration 2 respectively. Hence the training set for the *extended classifier* consists of the training instances of the base classifier and the training instances for the specialized classifier from iteration 1 or iteration 2 respectively.

Examining the results of CENFA for the S class, the AUC decreases first but finds a peak at a *dct* of 0.8. In the FT class, the AUC increases from a *dct* of 0.5 to 0.55 and stays on that level until it decreases sharply for *dct* over 0.8.For the R&D class, the *dct* between 0.5 and 0.65 leads to no improvement of the AUC and makes it decline for higher *dct*, though not as sharply as seen for the FT class. This shows that a reasonable *dct* has to be identified separately for each scenario as a general best value cannot be found.

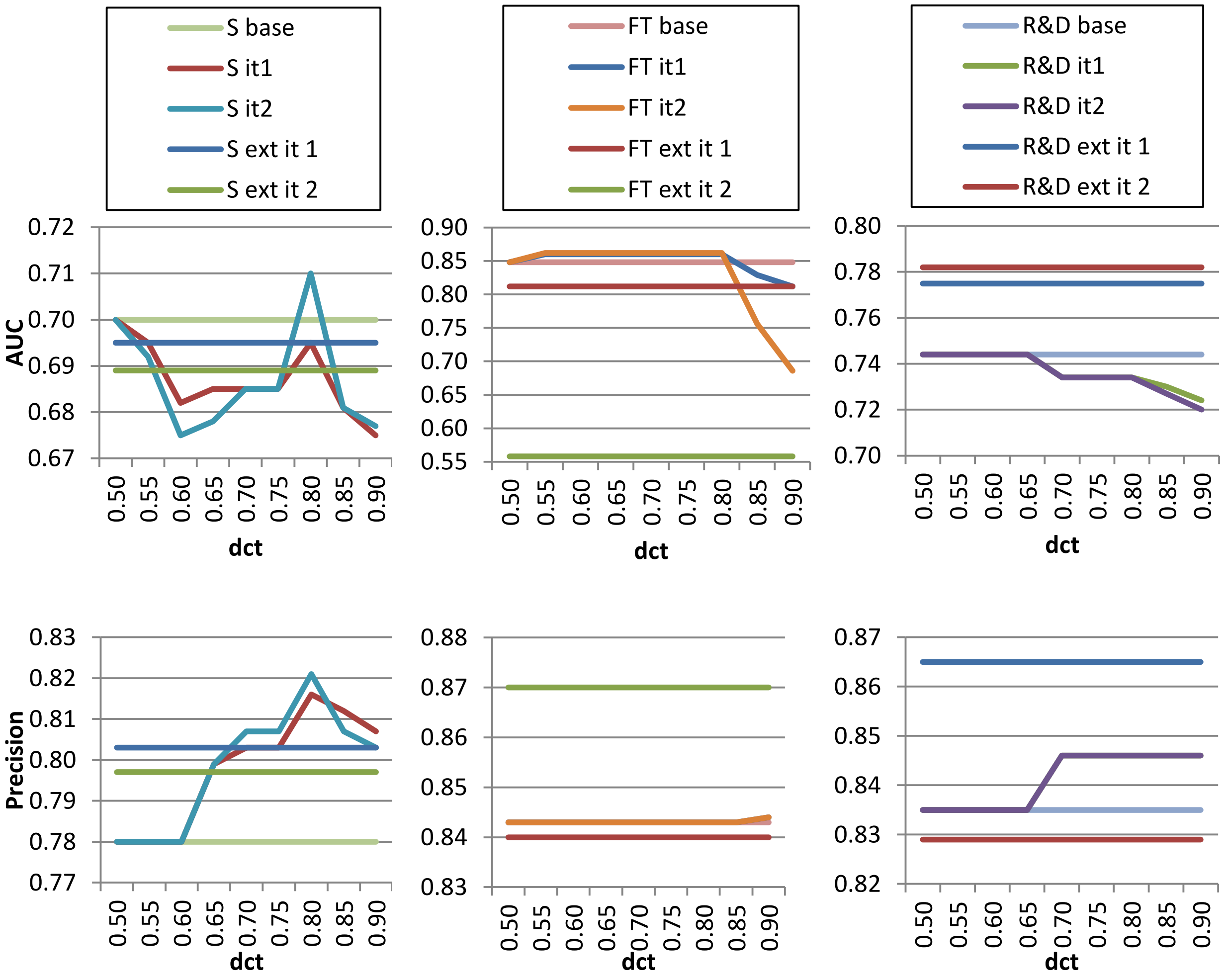

Figure 5: AUC and weighted average precision for the different classes with varying *dct* throughout the iterations (*act*=0.5). The y-axis are scaled differently because of the different magnitudes of change.

Analyzing the single classes in more detail and with regard to the precision of the model, the evaluation results show that the tradeoff differs between the different scenarios. One can see that the precision for the *S* class can be increased, depending on the value of the *dct* and that precision and AUC reach their peaks both at a *dct* of 0.8. After the second iteration we achieve here an increase in precision of 5.3% and an increase in AUC of 1.4% while for other *dct* values only the precision can be increased.
The figures for the *FT* class look different. Here the AUC is constantly increased by 0.012 and there is no impact on the precision until a *dct* greater than 0.8 is chosen. After that threshold, the AUC drops dramatically while the precision is improved a little. Looking at the results for *R&D*, the results for precision and AUC run more parallel than for the other classes. Here the iterations have no effect for a *dct* between 0.5 and 0.65 while AUC drops for higher *dct* and precision increases. Comparing the AUC of the base classifier (0 .74) with the average AUC for 180 training instances in the tuning experiments (0.83) (cf. Figure 2), one can observe a decreased value. This is assumed to result from the particular fold selected for the experiments in the industrial setting. The standard deviations in the tuning experiments (cf. Figure 2) show the variety for the different folds.

It was shown before [3] that in some cases the *extended classifier* outperforms the CENFA classifier in terms of classification quality while in some cases the CENFA classifier outperforms the *extended classifier*. This trend can be observed in this industrial setting as well. As presented before [3], CENFA provides a good trade-off between a good classification quality and a significantly better behaviour in terms of computation time compared to a pure ensemble learner such as the *extended classifier*.

Distribution of Ambiguous Documents

As mentioned in Section 4.2, the class distribution of the examined documents is highly unbalanced. Table 4 gives an overview on the distribution during the different phases of the approach's usage. The first row shows again the unbalanced results from the annotation study. The following rows show the results of the two post-annotation rounds during the iterations. Interestingly, the distribution becomes more balanced in these phases. One reason for this might be that the *act* represents an uncertainty window which spreads from the center of uncertainty equally into both classes. With respect to the applied classification algorithm of SVM, the model selects instances which are close to the separating SVMs' hyperplanes for the classes on either side and not randomly from the complete geometric space.

Table 4: Fraction of documents annotated as positive for the respective class during the different phases

Phase	*R&D*	*S*	*FT*
Evaluation Data	12.0%	16.7%	92.3%
1st Iteration	41.2%	42.3%	54.9%
2nd Iteration	28.4%	45.5%	48.3%

8 Conclusion

This work describes and evaluates a procedure for the creation of filters for domain-specific search engines in an industrial setting including an offline annotation study to train non-experts and applying an active-learning framework to use in an online evaluation.
The annotation study provides the description of a process to easily transfer the knowledge of a domain expert to non-experts for the annotation of complex filter scenarios. The high inter-

annotator agreement allows the conclusion that the collaborative, iteratively improved definition of the filters helped to create a common understanding of the problems. However, the differences in the agreement for the different problems show that for humans some filter problems remain more complex than others even after a thorough common definition.

Another result of the annotation study and the online evaluation is the created corpus, which is openly accessible and provides insights on the character of the identified ambiguous documents and enables repeated and comparative studies on the same data.

Furthermore, the previously presented concept of CENFA is adapted for the online setting and evaluated in a live-scenario with real data streams as a means for filtering of documents for a domain-specific search engine. The results give interesting insights into the suitability of the approach. One main contribution is the study on the impact of the different parameters on the classification results.

The article is completed by practical considerations such as time requirements for creation of filter definitions and human annotations that can help developers from industry to better calculate for the implementation of any text classification based filter.

An aspect that would be of interest for future work is the system's behavior during long term usage. It might be interesting to see at which point of time the classification results stabilize and a re-training of the base classifier rather than the specialized classifier would be beneficial. Furthermore, a qualitative analysis of the ambiguous documents might help to identify suitable instances for the initial training data annotation.

Acknowledgements

The work presented in this paper was partly funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IS12054 and partly funded by a research grant of the German Research Foundation (DFG) in the project "Design und Bewertung neuer Mechanismen für Crowdsourcing". We extend deep gratitude to kimeta GmbH, in particular Elena Neuschild and Sven Kloppenburg, for enabling us to carry out this work within a real application scenario and to permit the publication of the resulting corpus.

References

[1] Netcraft, LTD, August 2014 Web Server Survey, 2014.
[2] M. Bagdouri, W. Webber, D.D. Lewis, D.W. Oard, Towards Minimizing the Annotation Cost of Certified Text Classification, Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 989-998.
[3] S. Schnitzer, S. Schmidt, C. Rensing, B. Harriehausen-Mühlbauer, Combining Active and Ensemble Learning for Efficient Classification of Web Documents, Polibits, (2014) 39-45.
[4] A. Hanbury, M. Lupu, Toward a Model of Domain-specific Search, Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, Le Centre de Hautes Etudes Internationales d'Informatique Documentaire, Lisbon, Portugal, 2013, pp. 33-36.
[5] K.W. Wöber, Domain-specific Search Engines, Destination Recommendation Systems: Behavioural Foundations and Applications, CABI, 2006, pp. 208-213.
[6] S. Oyama, T. Kokubo, T. Ishida, Domain-specific Web Search with Keyword Spices, IEEE Transactions on Knowledge and Data Engineering, 16 (2004) 17-27.

[7] P. Sondhi, R. Chandrasekar, R. Rounthwaite, Using Query Context Models to Construct Topical Search Engines, Proceedings of the Third Symposium on Information Interaction in Context, ACM, New Brunswick, New Jersey, USA, 2010, pp. 75-84.
[8] A. Kruger, C.L. Giles, F.M. Coetzee, E. Glover, G.W. Flake, S. Lawrence, C. Omlin, DEADLINER: Building a New Niche Search Engine, Proceedings of the Ninth International Conference on Information and Knowledge Management, ACM, McLean, Virginia, USA, 2000, pp. 272-281.
[9] G. Bo, Y. Yichen, X. Chao, H. Xian-Sheng, Ranking Model Adaptation for Domain-Specific Search, IEEE Transactions on Knowledge and Data Engineering, 24 (2012) 745-758.
[10] Y. Fu, X. Zhu, B. Li, A Survey on Instance Selection for Active Learning, Knowl Inf Syst, 35 (2013) 249-283.
[11] D. Vasisht, A. Damianou, M. Varma, A. Kapoor, Active Learning for Sparse Bayesian Multilabel Classification, Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, New York, USA, 2014, pp. 472-481.
[12] S. Tong, D. Koller, Support Vector Machine Active Learning with Applications to Text Classification, The Journal of Machine Learning Research, 2 (2002) 45-66.
[13] T. Joachims, A Statistical Learning Model of Text Classification for Support Vector Machines, Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2001, pp. 128-136.
[14] Z. Lu, J. Bongard, Exploiting Multiple Classifier Types with Active Learning, Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, Montreal, Quebec, Canada, 2009, pp. 1905-1906.
[15] M. Sugiyama, N. Rubens, Active Learning with Model Selection in Linear Regression, Proceedings of the Eighth SIAM International Conference on Data Mining (SDM2008) 2008, pp. 518-529.
[16] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American, May 2001, pp. 28-37.
[17] J. Cohen, A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20 (1960) 37-46
[18] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, Experimental Perspectives on Learning from Imbalanced Data, Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 935-942.
[19] Y. Kim, Analysis of Query Entries of a Job Search Engine, in: A. Marcus (Ed.) Design, User Experience, and Usability. Web, Mobile, and Product Design, Springer Berlin Heidelberg 2013, pp. 203-211.
[20] Z. Lu, M. Bada, P. Ogren, K.B. Cohen, L. Hunter, Improving Biomedical Corpus Annotation Guidelines, Proceedings of the joint BioLink and 9th Bio-Ontologies Meeting, 2006, pp. 89-92.
[21] S. Ertekin, J. Huang, L. Bottou, L. Giles, Learning on the Border: Active Learning in Imbalanced Data Classification, Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management, ACM, 2007, pp. 127-136.
[22] N.V. Chawla, Data Mining for Imbalanced Datasets: An Overview, Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 853-867.
[23] S. Schnitzer, Effective Classification of Ambiguous Web Documents Incorporating Human Feedback Efficiently, Faculty of Computer Science, University of Applied Sciences Darmstadt, Darmstadt, Germany, 2013.
[24] E. Hovy, J. Lavid, Towards a ‘science’ of corpus annotation: a new methodological challenge for corpus linguistics, International Journal of Translation, 22.1 (2010) 13-36.
[25] Q. Gu, Z. Cai, L. Zhu, B. Huang, Data mining on imbalanced data sets, Proceeedings of the 2008 International Conference on Advanced Computer Theory and Engineering, IEEE, 2008, pp. 1020-1024.
[26] J. R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, Biometrics, 1977, pp. 159-174