Please quote as: Hoffmann, H.; Bullinger, A. & Fellbaum, C. (2013): Towards the automated evaluation of crowd work: Machine-learning based classification of complex texts simplified by laymen. In: 46th Hawaii International Conference on System Sciences (HICSS 2013), Grand Wailea.

Towards the automated evaluation of crowd work: Machine-learning based classification of complex texts simplified by laymen

Holger Hoffmann
Kassel University
hhoffman@uni-kassel.de

Angelika Bullinger
TU Chemnitz
awi@mb.tu-chemnitz.de

Christiane Fellbaum
Princeton University
fellbaum@princeton.edu

Abstract

The work paradigm of crowdsourcing holds huge potential for organizations by providing access to a large workforce. However, an increase of crowd work entails increasing effort to evaluate the quality of the submissions. As evaluations by experts are inefficient, time-consuming, expensive, and are not guaranteed to be effective, our paper presents a concept for an automated classification process for crowd work. Using the example of crowd generated patent transcripts we build on interdisciplinary research to present an approach to classifying them along two dimensions – correctness and readability. To achieve this, we identify and select text attributes from different disciplines as input for machine-learning classification algorithms and evaluate the suitability of three well regarded algorithms, Neural Networks, Support Vector Machines and k-Nearest Neighbor algorithms. Key findings are that the proposed classification approach is feasible and the SVM classifier performs best in our experiment.

1. Introduction

Already in his 2005 bestseller 'The wisdom of the crowd', Surowiecki has outlined the potential of a new work paradigm: broadcasting tasks to "smart people" outside the organization [1]. Meanwhile, this organizational concept of *crowdsourcing* has surged and found application in a wide number of areas and at different stages of value creation [2].

Crucial to the successful implementation of crowdsourcing is the use of an open call or broadcast search [3] and a large network of potential participants [2] in order to receive many submissions by the crowd to ultimately solve the problem [4]. Typically, submitted crowd work is evaluated by a jury of experts who are part of the sponsoring firm [5]. This manual procedure poses a dual challenge of efficiency and effectiveness, which is currently not answered [6].

First, we face a challenge of *efficiency*. Evaluation of crowd work can be very time-consuming due to the sheer number of submissions. In the example of Google's 10^100 project, 3,000 employees took part in the evaluation of submissions. Nonetheless, evaluation of the 150,000 submissions delayed the project for 24 months [7]. The question arises whether human experts are the most efficient alternative to assess crowd work.

Second, there is the challenge of *effectiveness*. Experts recruited from the sponsoring firm are subject to mental barriers, most important the "not invented here" syndrome [8]. It has also been shown that, on average, expert panels do not provide predictive power concerning the success of the evaluated product [9]. The question hence arises whether human experts are the most effective possibility to assess crowd work.

Our research [1]makes a contribution to the field of crowdsourcing by i) examining the problems of evaluation and ii) proposing a process to improve efficiency and effectiveness of evaluation. This process goes beyond current, human-based approaches to evaluation. It is based on machine-learning algorithms for the classification of crowd work, using attributes identified from literature that can be used as input for the machine-learning algorithms.

For our research, we focus on the evaluation of a crowdsourcing project: The PatViz project aims to improve accessibility of expert knowledge coded in nomenclature. Using the example of patent texts which require technical knowledge as well as fluency in the peculiar style of writing in combination with legal terms, informally known as 'legalese', to be assessed and understood, the PatViz approach follows the work by Shinmori et al. [10]. The approach by Shinmori et al. makes patent texts more accessible for legal laymen by replacing or

[1] Our work was supported by the Peter-Pribilla-Foundation and grants CNS 0855157 and CCF 0937139 from the U.S. National Science Foundation.

1530-1605/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2013.568

explaining complex terms in the original patent texts using words that are easier to understand. Going one step further in the PatViz project, a crowdsourcing effort generated transcripts from 55 patent texts into plain English, rewriting complete patent abstracts instead of only replacing single words or phrases.

The PatViz project decided to have these transcripts composed not by designated individuals who are experts in the field of the patent and also have a fair understanding of the legal terminology, but by legal laymen in a crowdsourcing approach. Thus, participants on Amazon Mechanical Turk were asked to generate the transcripts and to adapt the patent's use of language as well as its logical and narrative structure [11] to a more readable form. As a result of the crowdsourcing activity, 550 transcripts of patent texts have been collected. This approach is both cheaper and faster compared to have the transcripts written by experts. However, as is the nature of crowd work, these transcripts are of different accuracy and understandability and have to be evaluated. The traditional path to evaluation, i.e. the recruitment of experts to assess the transcripts generated by the crowd would exhaust the savings in time and money realized by employing laymen as transcribers in the first place.

Thus, we devised a machine-learning based classification approach for patent texts simplified by laymen. To achieve this, we build on current research in the fields of patent information retrieval, computer linguistics and machine-learning to determine transcript attributes that can be used for classification as well as evaluate different machine-learning approaches to perform this classification. In order to test the feasibility of our approach and evaluate the resulting classification, we conducted a set of experiments to test different classification approaches on the 550 transcripts of 55 patent texts that had been generated by the crowd on Amazon Mechanical Turk.

2. Related work

The research we are presenting in this article covers aspects in patent information retrieval and readability/reading level assignment using machine-learning approaches. Hence in this section, we give an overview over the body of related work that already exists and can be taken advantage of when combining the different approaches.

In current publications on *patent information retrieval* many different areas of interest are described. One major area of interest concerns different approaches to searching patent texts, e.g. high recall searches where as many as possible applicable patents are retrieved from a collection [12], and how those algorithms can be evaluated [13]. Another strand of research, that is highly relevant to our research, deals with classifying patent texts automatically. Benzineb and Guyot [14], e.g., employ machine-learning algorithms for assigning one or more categories to a patent that is not categorized in one of the given patent classification systems. Koster et al. [15] apply linguistic techniques to this task of classifying patent texts. Al Hasan and Spangler [16] also employ linguistic analysis, however their aim is to rank patents in a collection based on their novelty. Shinmori et al. [10] use language processing and even go one step further and attempt to improve the readability of patent texts automatically.

Readability assessment of texts is one of the major fields of research in linguistics and hence numerous different readability indices are described by literature. Among those used most often are the Flesch Reading Ease Scale, the Flesch-Kincaid Readability Formula, the Gunning Fog Index, SMOG Readability Formula, the Coleman-Liau Index and the Fry Readability Graph [11, 17-21]. As it would exceed the scope of this paper to compare all these approaches, we would like to highlight two indices mentioned in the literature that are very relevant for our approach. The *Flesch Reading Ease Scale* [22] measures the readability of text between grade 5 and college level. To calculate the readability, the underlying formula considers the average sentence length and average word length of a text. The *Gunning Fog Index* [23] measures the readability of texts between grade 4 and college level. In contrast to the Flesch Reading Ease Scale, the Gunning Fog index focuses on the average number of words per sentence and the percentage of words with three and more syllables in the text. As in the case of these two readability scores, such linguistic characteristics of the texts are the basis for readability assessment by the other approaches.

Heilman et al. [19] and Scarton et al. [21] both attempt to find an algorithmic readability assessment for texts. Heilman et al. approach this by evaluating different statistical models and features for automated reading difficulty prediction, while Scarton et al. employ a machine-learning approach for their readability assessment, using different linguistic attributes as input for the machine-learning approach. A slightly different approach, but also using a machine-learning algorithm, is presented by Meara, Rodgers and Jacobs [24]. They find assessments for text quality written by second-language learners.

While not directly concerned with readability the approach of using linguistic measures as an input for a neural network seems very applicable to our research.

3. An Automated Classification Approach

The main objective of our work is to present an approach that allows automating the classification of text-based crowd work into different classes. We base our research on existing research in related domains, most importantly machine learning for classification tasks as well as computer linguistics. As an example, we are using complex patent abstracts simplified by laymen – originating from the PatViz project – that are to be classified in two dimensions i) the correctness of the simplified text compared to the original text and ii) the ease of reading of the simplified texts, i.e. their readability.

In this section, we describe our approach and give an overview over the principles used in the individual process steps and illustrate them using the PatViz evaluation as case study. In our approach, we first gather the object for classification, the transcripts, define classes and determine a "gold standard" to evaluate the machine-learning approaches against (3.1). Parallel to that, we identify possible attributes used by the classifier (3.2) and select the most promising ones (3.3), both referencing extant literature. After this, we train different machine-learning algorithms for classification and use them to classify the crowd input (3.4). The final step is selecting a classification algorithm based on its classification quality (3.5 & 4).

3.1 Corpora, Classes and Gold Standard

In computer linguistics, the objects of analysis is referred to as the *corpus*. For our research aiming at classifying simplified patent abstracts, the sources for our corpora are patent texts and the laymen generated simplified texts. We obtained the text bodies considered in our case study from two sources: randomly selected patent abstracts from the United States Patent and Trademark Office and crowd generated patent abstract transcripts created by users on the Amazon Mechanical Turk Platform.

The *corpus of the patent abstracts* contains the text abstracts of 55 patents, randomly chosen from patents issued in December 2010 and covering topics related to Computer Science / Information Systems. The latter constraint is necessary due to our access to domain experts' subsequent generation of the reference classification of our corpora. The patent topic, as found in the International Patent Classification system, occurring most often is "Computing; Calculating; Counting" (n=39), followed by "Electric Communication Technique" (n=27), "Information Storage" (n=4) and "Signalling" (n=3). The *corpora of the patent transcripts* are comprised of ten crowd-generated transcripts of the original abstract per patent. We obtained the transcripts using the Amazon Mechanical Turk marketplace, offering a reward of 0.80 USD for participants to complete the task with the following description:

Formal technical texts are often hard to understand for the average reader. Please rewrite such a text in words that are easier to understand. Your abstract should include around 4-5 sentences.

The resulting 55 corpora of 10 texts each are the main subject for our research, aiming at classifying those crowd-generated texts based on their readability and their correctness in reproducing the patent abstracts contents. To establish a gold standard, i.e. the reference classification taken to be correct, we asked 8 domain experts to classify all 550 transcripts regarding the two dimensions we are interested in *–readability* and *correctness–* as very high, high, low or very low. The gold standard serves the purpose of a training set for the machine-learning algorithms as described in section 3.3. In order to ensure the reliability of the gold standard, we identified the outliers for both readability and correctness for each transcript corpus. To do this we applied Peirce's criterion, which is commonly used for outlier identification [25]. The average of the remaining expert ratings finally represent the gold standard classification used for the next steps in the evaluation process. Results of outlier identification and adaption of the classifications are consistent with a RSME analysis of the expert ratings. Figure 1 shows the gold standard resulting from the classification done by the experts.

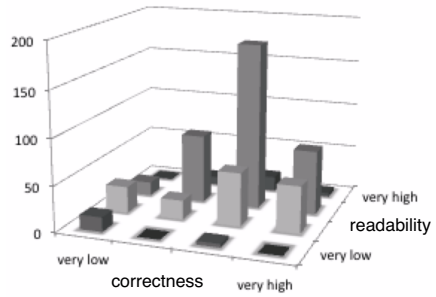

Figure 1: Expert classification (Gold Standard)

As we were expecting attempts to cheat on the task (see e.g. [16]) we also checked for unusual transcripts and found six transcripts by users that were overburdened by the task, writing i.e., "I really don't know" or "I don't understand the text", but only four real attempts to cheat, e.g. by copying the text instead of transcribing it or entering random text. We kept these texts in the copora, as such outliers are always to be expected in crowd-based projects and also need to be identified correctly in a machine-learning based classification approach.

3.2 Attribute Identification

In order to be able to automatically classify the transcripts, we need to identify and later select a set of attributes that represent different aspects of the texts. To describe the individual transcripts we can use attributes covering different aspects of the text. On one hand, we can use attributes describing the linguistic characteristics of the text, on the other hand we can use attributes that describe the relation between the transcript and the original patent abstract.

3.2.1 Linguistic attributes. For the *linguistic aspects* we rely on the widely used *Coh-Metrix* analysis tool (c.f. [11, 21, 26]) to cover all basic aspects of the texts. As the classification task at hand works on short transcripts of longer, more complex texts, we also include metrics calculated by the Recall-Oriented Understudy for Gisting Evaluation *(ROUGE)* evaluation tool, which is used to evaluate the quality of text summaries [27].

Coh-Metrix is a web-based tool for analyzing texts regarding cohesion relations, language used and readability [28]. The attributes Coh-Metrix calculates are divided into five categories [29]: General Word and Text Information, Readability Indices, Syntax Indices, Referential and Semantic Indices and the Situation Model Dimensions.

- *General Word and Text Information* includes attributes on word and text level, referencing e.g. the usage frequency of a word in the English language [29]. Examples for attributes include the number of words and sentences in the text, average number of words per sentence or the hypernym value to describe a word's abstractness (c.f. [30]).
- The *Readability Indices* describe how easy it is for a reader to understand a text. Coh-Metrix calculates two of the most common attributes, the Flesch Reading Ease and the Flesch-Kincaid Grade Level [29]. We additionally considered the Gunning Fog index [23], SMOG [20] and the Coleman Liau index [17] as alternative approaches to assess the readability of text often found in literature.
- *Syntax Indices* include attributes that assess the syntactic complexity and composition as well as the frequency of syntactic classes for a text [29]. Attributes are, e.g. incidence scores for noun-phrase constituents or negation expressions, rate of pronouns and number of additive, temporal, logical, or causal connectives in a sentence.
- *Referential and Semantic Indices* describe the cohesion within a text [29]. One example for an attribute is, e.g., how often a noun refers to another constituent in the text. Another attribute is the adjacent stem overlap, the ratio of neighboring sentences with one or more common word stems.
- The *Situation Model Dimensions* aim at reflecting the mental model the text spans up. Attributes for the Situation Model Dimensions cover the five situational dimensions causation, intentionality, time, space and protagonists [29].

Using the web-based tool hosted by the Department of Psychology at the University of Memphis, we calculated the values of 56 attributes for all 550 transcripts. Some preprocessing for the transcripts was necessary, as the texts were sometimes syntactically malformed, e.g., two periods following each other, and Coh-Metrix appeared to crash when trying to work with the "empty sentence" between such two periods. Except for removing these elements, no other alternations were carried out on the transcripts.

The second tool for text analysis, the *ROUGE* package, aims at evaluating the quality of computer-generated text summaries [27]. The quality of a summary is determined by ROUGE by comparing the summary to "ideal" summaries, usually created by humans. In this comparison between the computer generated summary and the ideal summaries, linguistic measures like the number of overlapping units, e.g. word pairs, are used. The main algorithms described by Lin [27] are based on *n-grams* (ROUGE-N, where n is the length of the n-gram), (weighted) *longest common substrings* (ROUGE-L, ROUGE-W) or *skip bigrams and unigrams* (ROUGE-S, ROUGE-SU). Overall they present 17 ROUGE scores: ROUGE-N with n = 1 to 9, ROUGE-L, ROUGE-W as well as ROUGE-S and ROUGE-SU with maximum skip distances of 0, 4, and 9.

We calculated all the ROUGE scores using the patent abstracts and the crowd transcripts as input. This allows us to evaluate whether one of these ROUGE scores can be used as an attribute for the machine-learning algorithms to classify the transcripts.

3.2.2 Relational attributes. Concerning the *relational aspects* between the transcript and the original abstract several possible attributes found in data clustering approaches can be used. The rationale behind this is, that a patent abstract and its transcripts form a logical cluster of texts belonging together. So in addition to linguistic measures based on single transcripts, we apply measures of *cluster cohesion* and *transcript separation* are taken from *data mining* research to determine the quality of clusters and the degree with which transcripts fit in their cluster [31]. As a result, we derive the three attributes from library sciences and data mining: individual *document distances, patent clusters' cohesions* and *transcript silhouettes.*

The *document distance* or Euclidean distance between two documents is a measure of dissimilarity between the two document vectors. It is derived from the cosine of the angle between the document vectors, found by incorporating the Euclidean norm of each document's frequency vector and the dot product of the two documents' vectors.

Once the distances between all the documents are known, the clusters' cohesions and the transcripts' silhouettes can be calculated. In this context, the *cohesion of a cluster* of documents is defined as the average distance between documents within the cluster. The cohesion thus is based on the distance function and shares the range in possible values [31].

The *silhouette* of a transcript is an attribute for how well it fits in its cluster compared to the other clusters. It hence puts the transcript's average distances to documents *within its own* cluster in relation to the transcript's minimum average distances to documents in *different* clusters [31, 32].

3.3 Attribute Selection

After identifying the 74 potential attributes from the literature and calculating the values for these attributes for the corpora, the attributes to be used by the classification algorithms have to be selected. In doing so we aim at removing redundant attributes (e.g. if one attribute can be expressed using one ore more other attributes) and irrelevant attributes that only reduce the accuracy of the classifications by increasing the dimensionality of the problem set [33]. For our research we use the *InfoGainAttributeEval* and the *SVMAttributeEval* evaluators of the WEKA software package [34].

The *InfoGainAttributeEval* algorithm determines the importance of an attribute by determining the information gain by this attribute with respect to the class, it was chosen for our work due to its popularity in the relevant literature, e.g. [21]. As we aim at classifying transcripts in two dimensions, correctness and readability, we ran the filter once for each dimension. For the "correctness" class, the algorithm kept 42 attributes that do contribute to information gain, for the "readability" class, it kept 44 attributes. The table below lists the top ranked attributes. Two things are especially noteworthy here: the dominance of the different ROUGE algorithms and the inclusion of two readability attributes for the readability class, while none is chosen for the correctness class.

Table 1: Highest ranked attributes (InfoGain)

Correctness	Readability
ROUGE-SU	Document distance
ROUGE-S	ROUGE-2
ROUGE-L	ROUGE-S
ROUGE-W	ROUGE-SU
ROUGE-2	ROUGE-L
Document distance	ROUGE-W
Document silhouette	Document silhouette
Number of words	Number of words
Type-Token ratio	Flesch reading ease
Pronouns-noun ratio	Gunning fog score

The *SVMAttributeEval* algorithm based on [35] determines the importance of an attribute by using the output of a classifier based on Support Vector Machines (SVM, see next section). This alternative approach was chosen to select attributes in a way that allows a better comparison of the SVM-based classification algorithm with other types of classifiers later in the process. As *SVMAttributeEval* does not identify attributes that don't contribute to the goal of the classification algorithm, but simply gives a ranked list of attributes, we have selected the same number of attributes the InfoGain algorithm identified as relevant – i.e. 42 for correctness and 44 for readability – to minimize (dis)advantages between the two selection algorithms based on the number of attributes used. Consistently with the theoretical background, both algorithms also rank readability attributes (Flesch reading ease, Gunning fog, SMOG and Coleman Liau) among the most relevant attributes for readability classification. For the classification of correctness though, these scores

show only minor influence. It is also noteworthy that the different ROUGE scores dominate the attribute rankings and are the top ranked attributes for both classification dimensions when using the InfoGain algorithm.

Table 2: Highest ranked attributes (SVM)

Correctness	Readability
Neg. causal connectives	ROUGE-4
ROUGE-S	Flesch reading ease
ROUGE-L	ROUGE-L
ROUGE-SU	ROUGE-S
Log freq. of content words	Sentence syntax similarity
ROUGE-4	ROUGE-SU
Ratio causal particles to causal verbs	Log min. raw freq. of content words
Number of words	SMOG
Document distance	Coleman Liau
No of noun-phrase const.	Number of words

3.4 Classification Algorithms

Current literature describes a plethora of possible machine-learning algorithms for classification, reviews of most approaches can be found in [24, 36]. For the scope of our research we limit ourselves to three approaches very frequently used for automatic classification [37, 38]: Artificial Neural Networks (NN), Support Vector Machines (SVM) and the *k* Nearest Neighbors (kNN) approach.

An artificial *Neural Network* is a network comprised of interconnected nodes, also called neurons, organized in layers [39, 40]. There are typically multiple nodes in the *input layer* –which in our case represent the different attributes– one or more *hidden layers* –nodes in those layers are used only internally with no distinct meaning– and one or more nodes in the *output layer* –in our case one node representing either the correctness or the readability. Each artificial neuron reads the values of its input neurons, weighs and aggregates them before using the result as input for its *activation function*, which returns the neuron's output value. Learning is achieved by *backpropagation*, where the weighs used by the individual neurons are adapted so that the network's output value(s) for a set of training input come closer to the given correct output value(s) in the training set. The type of artificial Neural Network we are using is a *Multilayer Perceptron (MLP)*, a network that maps the input data forward to the output node in a directed graph.

In contrast to this, the *Support Vector Machines'* underlying principle is to take a set of training data as an input and find hyperplanes separating the known classes in the multidimensional representation of the data [41]. In our case, each of the attributes is represented in one dimension, i.e. a SVM classifier for two attributes would find lines separating the clusters in two-dimensional space, for three attributes it would find planes separating the clusters in three-dimensional space and so on. When finding a hyperplane that separates clusters, the optimal solution is the hyperplane that generates the maximum margins between itself and the classes [41], as this ensures the best results after training (cf. Figure 2).

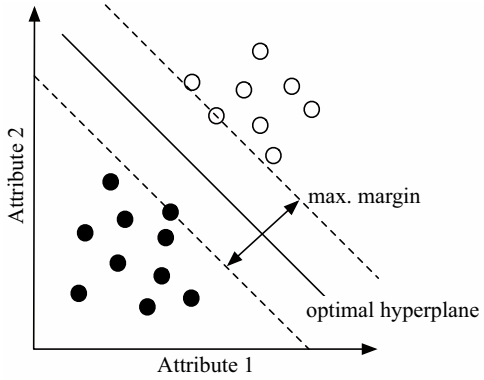

Figure 2: Example for an optimal hyperplane [41]

The *k Nearest Neighbors* approach is one of the simplest classification algorithms [42]. Similar to Support Vector Machines, the individual attributes span a multidimensional representation with the training set populating the space. Elements are classified, based on their relative distances in this multidimensional space to members of the respective classes, taking the k nearest neighbors into account. So given two classes A and B, a new element e would be classified as belonging to A by a 5 Nearest Neighbor classifier if at least three of its nearest neighbors are classified as A. The choice of an appropriate number for k is vital for the quality of the classifier, as noted by [42]. As [21] tested values from 1 to 10 for their two-class problem and found that k=7 resulted in the best classifications we tested an interval of 1 to 60 for our problem using the quality criteria below and found *k=10* to deliver the best classification results.

Prior to putting these algorithms to use, they have to be properly trained. For Neural Networks this is done using the backpropagation approach described above, for the other algorithms the set of training data creates the multidimensional working

space and is used to find the separating hyperplanes (SVMs) or as reference points (kNNs). One common technique to train machine-learning algorithms is to use a subset of elements from the gold standard, [36] mentions a sample size of 2/3, as training set for the algorithms. Another option is cross-validation, where the input set is partitioned into mutually exclusive and equally-sized subsets of data and the algorithm is trained on each of these subsets and the remaining subsets are used validation data [36]. As we try to reduce the expert effort for classifying large datasets, we are looking for algorithms that perform well using only little expert input as training data. We hence use a *10-fold cross validation* approach for training our algorithms, meaning that we check the algorithm quality with $1/10^{th}$ of the dataset as training input. In a real world example this means that 10% of the crowd work would have to be classified by experts and used as training data for the machine-learning classifier before attempting to classify the other 90% without expert involvement. Using the cross-validation method also ensures that the results are not biased by (un)fortunate random selections from the gold standard as training data.

3.5 Algorithm Selection & Quality Criteria

The selection of classification algorithms for a classification problem is usually based on the accuracy of the classifier, i.e. what percentage of elements are classified correctly [36]. After training the algorithms as described before, i.e. by taking a sample of the gold standard as a training set, the remaining subset from the gold standard –called the verification set– is classified by the algorithms. The results of the individual classifications can then be compared to the known classification in the verification set. To compare algorithms based on these results, the fields of information retrieval and pattern recognition use the *precision,* the *recall,* and the *f-score* of an algorithm. The values of all three measures lie in an interval between 0 (worst) and 1 (best). This can directly be applied to the classification problem. For classification, the *precision* is defined as the number of items in a class that are classified correctly divided by the total number of items classified as members of that class. *Recall* on the other hand is defined as the number of items in a class that are classified correctly divided by the total number of items that actually are members of that class. The *f-score* combines both the precision and the recall of a classification and represents a weighted average of those values.

An example: our algorithm predicted 5 elements as members of class A, of which 3 are actual members of class A while 2 are members of class B. Also, in reality, class A contains 6 elements. Here, our algorithm's precision for class A equals 3/5 = 0.6, the recall 3/6 = 0.5 and the f-score is about 0.55.

Another approach for determining the quality of classification approaches also mention *Receiver Operating Characteristic (ROC) curves*, a representation of the interrelation between the approach's *true positive rate* and *false positive rate* [43]. The *area under the curve (AUC)* is then used as a quality measure, with a value of 1 for a perfect classifier and 0.5 for random classifications [44]. To evaluate the results of our case study, we will hence use precision and recall information in combination with ROC curves and the rate of correctly classified texts to compare the different algorithms.

4. Results of the classification experiment

In order to determine which machine-learning algorithm is most suitable for the task of classifying the crowd transcripts and to evaluate the effect of filtering out attributes we evaluated all three algorithms for individually for both dimensions and used the complete set of attributes identified in the literature (c.f. Table 3) as well as the attribute sets filtered using the InfoGain (c.f. Table 4) and SVM (c.f. Table 5) algorithms. As quality criteria we used the *accuracy of the classification* (*acc.*), i.e. the percentage of correctly classified transcripts, the *f-score* (*f-sc.*) combining the classifier's recall and precision, and the *area under the ROC curve* (*AUC*, c.f. Figure 3).

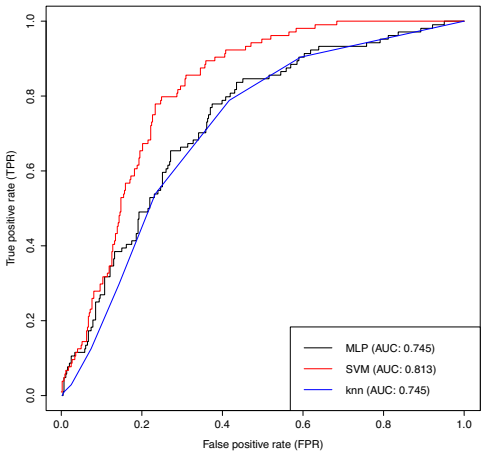

Figure 3: ROC Curves for classifiers using SVM filtered input to find transcripts' correctness

In all three scenarios, the SVM algorithm achieved the best classification results concerning the

dimension of transcript correctness. The same is true for the classification of transcript readability when using the raw input. But when filtering out irrelevant attributes, the performance of the 10-Nearest Neighbor comparison matches and slightly outperforms the results of the SVM algorithm. The Multilayer Perceptron did not generate the best classification result in any of the scenarios or for any of the dimensions.

As expected, performing the attribute selection using the InfoGainAttributeEval algorithm and the SVMAttributeEval algorithm did indeed lead to better classification results for the AUC measure in both dimensions and for all classification algorithms. However, for the two other measures this improvement cannot be seen. When filtering out attributes using the InfoGain algorithm, accuracy and f-scores do not change much, with the exception of accuracy dropping for the SVM classifying readability. Using the SVMAttributeEval algorithm to filter out slightly improves the SVM classification of correctness while its classification of readability drops considerably.

Table 3: Classification quality of unfiltered input

	Correctness			Readability		
	Acc.	**f-sc.**	**AUC**	**Acc.**	**f-sc.**	**AUC**
MLP	0.564	0.563	0.730	0.635	0.628	0.694
SVM	**0.615**	**0.609**	**0.792**	**0.713**	**0.683**	**0.710**
kNN	0.565	0.538	0.722	0.695	0.654	0.690

Table 4: Classification quality after InfoGain filter

	Correctness			Readability		
	Acc.	**f-sc.**	**AUC**	**Acc.**	**f-sc.**	**AUC**
MLP	0.538	0.537	0.732	0.636	0.635	0.692
SVM	**0.615**	**0.605**	**0.804**	0.675	**0.654**	**0.727**
kNN	0.595	0.570	0.748	**0.698**	0.652	0.713

Table 5: Classification quality after SVM filter

	Correctness			Readability		
	Acc.	**f-sc.**	**AUC**	**Acc.**	**f-sc.**	**AUC**
MLP	0.567	0.565	0.745	0.635	0.629	0.714
SVM	**0.620**	**0.611**	**0.813**	0.665	0.652	**0.734**
kNN	0.553	0.530	0.745	**0.704**	**0.665**	0.721

Comparing the measures for all algorithms to other examples in the literature, they would be judged as no better than fair for the Multilayer Perceptron and the 10-NN algorithm and just barely good for the SVM algorithm. However, most of the classification tasks found in literature only consider two classes (e.g. true and false), while we are working with four classes (very low, low, high, very high) per dimension. Hence we consider accuracy levels well above the guessing probability (25%), an f-score above 0.6 and an AUC over 0.8 as very satisfactory results for our experiment. This assessment is shared by potential users in practice: we have been invited to discuss the possible integration of our approach in a tool for national and trans-national patent offices.

5. Discussion and Future Research

In summary, the research presented in this paper comprised development and testing of an automated approach to improve efficiency and effectiveness of evaluation. The process is based on extant knowledge in linguistics and machine-learning which we perused to identify possible attributes as input for machine-learning based classification algorithms, to select the most relevant attributes and to evaluate the classification quality of different machine-learning based classifiers. Using the case example of patent texts, we have demonstrated the potential of an algorithm-based approach to evaluate large sets of crowd work. Since our approach is based on fundamental linguistic and machine learning principles, it can easily be generalized for use on texts in other domains.

As indicated for related classification tasks by extant literature, e.g. [38], our research showed that Support Vector Machines performed very well in all cases and are a good choice for solving our initial problem. Since it was part of our research to compare the applicability and performance of different algorithmic classification approaches, and since the SVMs we trained already outperformed the other algorithms, we did not focus on optimizing the SVMs. Concerning future research, we judge it appropriate to further optimize the SVMs used for classification, following [45], before applying such a classifier in a real world scenario.

From the conclusions drawn in [27], we expected to use one of the attributes calculated using either ROUGE-2, ROUGE-L, ROUGE-W or ROUGE-S as these algorithms worked well in single document summarization tasks. As ROUGE-1, ROUGE-L, ROUGE-W and ROUGE-SU algorithms also performed very well evaluating very short summaries, we also expected that one of them might provide a suitable input for our machine-learning algorithm. This was confirmed when using the InfoGainAttribute filtering algorithm, which ranked scores from ROUGE-2, ROUGE-L, ROUGE-S,

ROUGE-SU and ROUGE-W at the very top for the dimension of correctness as well as the readability. However, it was unexpected that the different ROUGE algorithms would be ranked so highly by both filtering algorithms.

From the logic behind input attribute filtering laid out in existing studies, e.g. [33], we did not anticipate that classifier quality was sometimes worse when using the filtered attribute input compared to classifier quality when using the complete set of attributes. This surprising result might be explained by the complete attribute set, that also includes attributes irrelevant for the classification task, leading to overfitting of the machine-learning algorithms, i.e. the classifiers adjusted to relevant attributes as well as to random, irrelevant attributes (i.e. noise) during the learning phase [40]. Even though this effect was rather weak in our experiment, future research could try to develop a more nuanced understanding of the relationship between the set of attributes and classifier quality and test whether the effect is related to overfitting or if there are other causes explaining the unexpected quality degradation.

In the empirical setup of crowd work, one finding concerning the results of the InfoGainAttributeEval algorithm is particularly interesting. When we tested the InfoGain attribute evaluation algorithm, it barely recognized a gain from the attribute that represents the rating of complexity and understandability by the person who wrote the transcript. As the algorithm found that the creators' input leads to very little information gain when establishing classification, the practice of self-rating found in some open innovation platforms appears not to be useful for highly complex tasks. In this area, our research holds important implications for both research and practice in the field. The current tendency to establish a classification of crowd work by a combination of self-assessment and peer evaluation (advocated among others by [46]) has to be reconsidered in the light of our findings. Related to this insight concerning self-assessment, future research should also determine whether crowd-rating, i.e., asking crowd-workers to rate the results of their peers, may be a source for additional features that could be used as input for the classification algorithms and help improve the classification result.

Finally, one aspect of crowd work was omitted when preparing the data for the classification algorithms so far. We decided not to filter out cheating attempts, but train and evaluate the machine-learning algorithms with the complete set of crowd work as it resulted from Amazon Mechanical Turk. As a next step towards a stable classification approach, future research should explore the effects of filtering out crowd work by overburdened users and cheaters. From our research we expect the filtering itself to be well feasible by using the attributes we already identified. For instance, we were already able to identify some types of worthless results from crowd-work that could be eliminated before an automated classification: first, (too) short transcripts by overburdened users and cheaters have a significantly smaller document norm compared to other transcripts; second, transcripts that are identical to the original text have a minimal dissimilarity; and third, transcripts that are completely unrelated show a large dissimilarity to the original text and true transcripts. To further refine the automated process to evaluate crowd work future research will need to evaluate the effect of removing those outliers before attempting to perform a classification and if there is a positive effect identify other attributes to filter out such foils.

References

[1] Surowiecki, J., *The Wisdom of Crowds.* 2005, New York: Anchor Books.

[2] Howe, J., *Crowdsourcing*. 2008, New York: Crown Publishing Group.

[3] Jeppesen, L.B. and K.R. Lakhani, *Marginality and Problem-Solving Effectiveness in Broadcast Search.* Organization Science, 2010. **21**(5): p. 1016-1033.

[4] Bullinger, A.C., J. Haller, and K.M. Moeslein, *Innovation Mobs*, in *15th Americas Conf. on Information Systems (AMCIS)*. 2009: San Francisco.

[5] Moeslein, K.M., J. Haller, and A.C. Bullinger, *Open Evaluation.* HMD, 2010. **273**: p. 21-34.

[6] Schulze, T., et al., *Idea Assessment in Open Innovation*, in *20th European Conf. on Information Systems*. 2012: Barcelona.

[7] Twohill, L. *$10 million for Project 10^100 winners*. Googleblog 2010 [cited 2010 Oct. 10th]; Available from: http://googleblog.blogspot.com/2010/09/10-million-for-project-10100-winners.html.

[8] Katz, R. and T.J. Allen, *Investigating the Not Invented Here (NIH) syndrome.* R&D Management, 1982. **12**(1): p. 7-19.

[9] Galbraith, C.S., et al., *Review panel consensus and post-decision technology performance.* Journal of Technology Transfer, 2010. **35**: p. 253-281.

[10] Shinmori, A., et al., *Patent Claim Processing for Readability*, Proceedings of the *Workshop on Patent Corpus Processing*. 2003, Association for Computational Linguistics. p. 56-65.

[11] Newbold, N. and L. Gillam, *The linguistics of readability.* Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing, 2010: p. 65-72.

[12] Bache, R., *Measuring and Improving Acceess to the Corpus*, in *Current Challenges in Patent Information*

Retrieval, M. Lupu, et al., Editors. 2011, Springer: Berlin. p. 147-165.

[13] Carterette, B. and E.M. Voorhees, *Overview of Information Retrieval Evaluation*, in *Current Challenges in Patent Information Retrieval*, M. Lupu, et al., Editors. 2011, Springer: Berlin. p. 69-85.

[14] Witten, I.H., E. Frank, and M.A. Hall, eds. *Data Mining*. Pratical Machine Learning Tools and Techniques. 2011, Morgan Kaufmann Publishers: Burlington.

[15] Koster, C.H.A., et al., *Phrase-Based Document Categorization*, in *Current Challenges in Patent Information Retrieval*, M. Lupu, et al., Editors. 2011, Springer: Berlin. p. 263-286.

[16] Al Hasan, M. and W.S. Spangler, *Assessing Patent Value through Advanced Text Analytics*, in *11th Int. Conf. on Artificial Intelligence and Law*. 2007: Stanford.

[17] Coleman, M. and T.L. Liau, *A computer readability formula designed for machine scoring.* Journal of Applied Psychology, 1975. **60**(2): p. 283-284.

[18] Friedman, D.B. and L. Hoffman-Goetz, *A Systematic Review of Readability and Comprehension Instruments Used for Print and Web-Based Cancer Information.* Health Education & Behavior, 2006. **33**(3): p. 352-373.

[19] Heilman, M., K. Collins-Thompson, and M. Eskenazi, *An analysis of statistical models and features for reading difficulty prediction*, in *EANL '08*. 2008: Stroudsburg.

[20] McLaughlin, G.H., *SMOG Grading - a New Readability Formula.* Journal of Reading, 1969. **12**(8): p. 639-646.

[21] Scarton, C., C. Gasperin, and S. Aluisio, *Revisiting the readability assessment of texts in Portuguese.* Advances in Artificial Intelligence–IBERAMIA 2010, 2010(LNAI 6433): p. 306-315.

[22] Flesch, R., *A new Readability Yardstick.* Journal of Applied Psychology, 1948. **32**(3): p. 221-233.

[23] Gunning, R., *The technique of clear writing*. 1952, New York: McGraw-Hill International.

[24] Meara, P., C. Rodgers, and G. Jacobs, *Vocabulary and neural networks in the computational assessment of texts written by second-language learners.* System, 2000. **28**.

[25] Ross, S.M., *Peirce's criterion for the elimination of suspect experimental data.* Journal of Engineering Technology, 2003.

[26] Hall, C., et al., *Using Coh-Metrix to Assess Differences between English Language Varieties*. 2007, University of Arizona Working Papers in Linguistics. p. 40-45.

[27] Lin, C.-Y., *Rouge: A package for automatic evaluation of summaries*, in *Proceedings of the Workshop on Text Summarization Branches Out*. 2004.

[28] Graesser, A., D. McNamara, and M. Louwerse, *Coh-Metrix: Analysis of text on cohesion and language.* Behavior Research Methods, Instruments & Computers, 2004. **36**(2): p. 193-202.

[29] *Coh-Metrix version 2.0 indices*. Available from: http://cohmetrix.memphis.edu/CohMetrixWeb2/HelpFile2.htm.

[30] Fellbaum, C., *WordNet: An Electronic Lexical Database*. 1998, Cambridge: MIT Press.

[31] Tan, P.-N., M. Steinbach, and V. Kumar, *Introduction to Data Mining*. 2005: Addison Wesley.

[32] Rousseeuw, P.J., *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis.* Computational and Applied MAthematics, 1987. **20**: p. 53-65.

[33] Lam, S.L.Y. and D.L. Lee, *Feature reduction for neural network based text categorization*, in *Proceedings of the 6th Int. Conf. on Database Systems for Advanced Applications* 1999. p. 195-202.

[34] Hall, M., et al., *The WEKA Data Mining Software: An Update.* SIGKDD Explorations, 2009. **11**(1): p. 10-18.

[35] Guyon, I., et al., *Gene selection for cancer classification using support vector machines.* Machine Learning, 2002. **46**: p. 389-422.

[36] Kotsiantis, S.B., *Supervised Machine Learning: A Review of Classification Techniques*, in *Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval*, I. Maglogiannis, et al., Editors. 2007, IOS Press: Amsterdam. p. 3-24.

[37] Benzineb, K. and J. Guyot, *Automated Patent Classification*, in *Current Challenges in Patent Information Retrieval*, M. Lupu, et al., Editors. 2011, Springer: Berlin. p. 239-261.

[38] Meyer, D., F. Leisch, and K. Hornik, *The support vector machine under test.* Neurocomputing, 2003. **55**: p. 169-186.

[39] Sarle, W.S., *Neural networks and statistical models*, in *19th Annual SAS Users Group Int. Conf.*. 1994.

[40] Tetko, I.V., D.J. Livingstone, and A.I. Luik, *Neural Network Studies. 1. Comparison of Overfitting and Overtraining.* Journal of Chemical Information and Modeling, 1995. **35**(5): p. 826-833.

[41] Cortes, C. and V. Vapnik, *Support-Vector Networks.* Machine Learning, 1995. **20**: p. 273-297.

[42] Hall, P., B.U. Park, and R.J. Samworth, *Choice of Neighbor Order in Nearest-Neighbot Classification.* The Annals of Statistics, 2008. **36**(5): p. 2135-2152.

[43] Fawcett, T., *An introduction to ROC analysis.* Pattern Recognition Letters, 2005. **27**: p. 861-874.

[44] Powers, D.M.W., *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation.* Journal of Machine Learning Technologies, 2011. **2**(1): p. 37-63.

[45] Eitrich, T. and B. Lang, *Efficient optimization of support vector machine learning parameters for unbalanced datasets.* Journal of Computational and Applied Mathematics, 2006. **196**(2): p. 425-436.

[46] Riedl, C., et al., *Rating scales for collective intelligence in innovation communities*, in *31st Int. Conf. on Information Systems*. 2010: Saint Louis.