Active Subtopic Detection in Multitopic Data

Benjamin Bergner
benjamin.bergner@st.ovgu.de

Georg Krempl
georg.krempl@ovgu.de

Knowledge Management & Discovery, Otto von Guericke University Magdeburg, Germany

Abstract

Subtopic detection is a useful text information retrieval tool to create relations between documents and to add descriptive information to them. For the task of detecting subtopics *with user guidance*, clustering by intent (CBI) has recently been proposed. However, this approach is limited to single-topic environments. We extend this approach for interactive subtopic detection in multi-topic environments, and for the incorporation of positive and *negative* user feedback. Our multi-topic clustering by intent (MCBI) approach iteratively constructs so-called similarity sets of documents within the same topic, derives candidates for new subtopics and actively queries feedback from the user, which is then used to refine the subtopic and similarity sets in the next iteration. For evaluation, we construct a corpus of the Wikipedia articles for the 4309 most common English nouns, comprising a broad range of different topics. Our MCBI approach is compared with the recently proposed CBI approach and random sampling. Each approach is evaluated based on the number of subtopics that are found in the same predefined, closed topic (countries). The results show that MCBI finds up to 137% and 445% percent more correct subtopics than random term selection or CBI, respectively.

1 Introduction

A business user might intend to group text documents, such as support logs or customer reviews, into meaningful subsets that correspond to the different subtopics addressed in the texts. Or, as further illustrative example, consider a user wants to cluster documents by using the different types of sport as subtopics. As pointed out in [7], it might be too tedious or even impossible to provide a priori an exhaustive list of possible subtopics, to label documents manually, or to experiment with different parameters until the desired clustering is obtained. However, it is comparatively easy for the user to illustrate the intended clustering by providing one (or a few) exemplary subtopics, for example by giving a cluster of tennis-related documents. Given the documents and initial subtopic(s), the clustering algorithm's task is to construct candidates for new subtopics, to actively seek the user's confirmation or rejection of these candidates as members of the intended topic, and to extend the clustering accordingly. Thus, this problem, recently described as *clustering by intent* in [7], is a combination of an active, incremental clustering task with very weak, interactive supervision.

A first approach for this task is provided in [7]. However, this approach is limited to single-topic environments. That is, documents unrelated to the intended topic might confuse the approach, like for example food-related

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

documents when the intent is deriving animal-related subtopics. Furthermore, the approach is focused on the user's confirmation of candidates, neglecting negative feedback by the user.

In this paper, we extend this approach to the interactive subtopic detection in multi-topic environments, and for incorporating positive and *negative* user feedback. Our **M**ulti-topic **C**lustering **B**y **I**ntent (MCBI) approach iteratively constructs so-called similarity sets, which comprise solely documents of the same topic. Then, it derives candidates for new subtopics, for which it actively queries feedback from the user. Information on the confirmed as well as on the rejected subtopics is then incorporated into the clustering model and used in subsequent iterations.

The remainder of this paper is structured as follows: in the next Section 2, we provide a more detailed background of clustering by intent and review the related work. Our method is presented in Section 3, followed by the experimental evaluation in Section 4. The paper closes with concluding remarks and an outlook to future work in Section 5.

2 Background and Related Work

We first provide a more formal definition of clustering by intent, in order to facilitate the subsequent discussion of the related work. Clustering by intent [7] corresponds to an interactive clustering task, where a given set of documents D should be partitioned into clusters. Each document $d \in D$ is represented by its bag of word feature vector $\vec{w}$. Each cluster c_{st} is defined and labelled by its corresponding subtopic st. The clusterer has neither access to an explicitly given similarity function, nor to an expert willing to perform a tedious tuning of such a function, nor to single similarity measurements between objects. Instead, the clusterer is provided with one (or few) exemplary clusters as an indication of the human *intent*. Its task is then to actively propose the user further clusters that should correspond to the same intent. Documents belonging to clusters that the user has already confirmed constitute the labeled set $L \subseteq D$. The unlabelled set $U \subseteq D$ comprises all remaining documents. This is illustrated in Fig. 1 a, where the two subtopics c_{Dog} and c_{Cat} as well as their corresponding documents (blue and yellow dots) have already been identified, while other documents (black dots) remain unlabelled. The approach proposed in [7] for CBI iteratively (1) trains a probabilistic multi-class classifier that discriminates between the known subtopics, (2) computes for each unlabelled document its confidence as the margin between its highest and second highest posterior estimate, (3) builds a so-called residual set that comprises the low-confidence documents, (4) selects the terms with highest precision in separating L from U as subtopic candidates, (5) queries their membership to the intent from the user, updates the sets and restarts the process again. This is illustrated in Fig. 1 b, where the most uncertain documents are used to form a residual set. The pseudocode for our implementation of this clustering by intent algorithm is given in Fig. 5 in the appendix. This clustering by intent is related to areas in constraint clustering and active learning, as well as to novel class detection, subgroup discovery, and topic detection, for which we will briefly review the most related literature below.

Within the rich literature on clustering, constraint clustering is of particular relevance. A recent survey on constrained (also denoted semi-supervised) clustering is provided in [5]. As an example, text clustering is given, where the objective is to group texts according to similarities in e.g. their content or their authors. The expert might provide must-link constraints for documents that are deemed to be similar, or cannot-link constraints for documents that are different. In [5], three categories are named, search based (constraint-based), where the solution space is modified e.g. by penalizing violations of constraints, distance based (similarity based), where the similarity function is modified to consider the constraints, or by hybrids of both. Active approaches, e.g. for selection of informative document pairs for which user feedback is queried [1], and more recently also noisy constraints have been researched [18]. However, in contrast to [7, page 33], constrained clustering assumes the similarity measure to be a priori specified, and the pairwise constraints require to query information on the document level, rather than on the subtopic level. In (inter)active clustering (e.g. [9, 6, 11]), not all similarities (or distances) are known a priori. Rather, the approach actively selects similarity measurements and queries them from an oracle. Recent works comprise the so-called interactive (hierarchical) clustering proposed in [11] and the active hierarchical clustering in [6]. However, these (inter)active clustering approaches require the similarity measure to be known a priori or similarity measurements to be provided by the user.

Many works in active learning literature addresses classification, where labels for instances are queried from the user [15]. Although some of these works address active learning on text data (e.g. sentiment classification in [10]), they assume the set of labels to be known and are therefore not applicable here. Similarly, in [14], active learning in text categorization is studied. The authors propose a two-fold process, where active learning is done

both on the document-level (querying document's category, i.e. its label), and on the term level. The latter corresponds to asking the oracle about the importance of features, i.e. asking for the most predictive words. However, the text categorization problem is posed as one-versus-the-rest classification problem, where the single category of interest is initially known. In contrast, in clustering by intent, the categories are not known a priori but rather need to be learned on the way. Nevertheless, the approach in [14] might be used as post-processing for labelling all documents, once the set of categories has been determined by a clustering-by-intent approach.

Another related field of supervised machine learning research is subgroup discovery ([17], see e.g. [2] for a recent survey). Given a population of individuals, its objective is finding as large as possible subgroups that show a distributional unusualness with respect to a certain property of interest [17]. The interestingness of a pattern is measured by a quality function [2], which is in general exploiting the target concepts' distribution. A common exemplary quality function is to combine a measure for the subgroup's size with its from the whole population in the target concept value. There exist interactive subgroup discovery frameworks (e.g. [8] that allow the expert to affect the attributes that are used for learning. However, the quality function and the target concept (i.e. class attribute) need to be specified a priori.

Furthermore, novel class detection (e.g. [12]) in data streams is a related area of research. There, the objective is to detect novel classes, whose instances differ from those of already known classes. Some approaches are passive (e.g. [12]), while others actively query labels from the user (e.g. [13, 3]). However, similar to clustering these approaches rely on an a priori specified similarity function. Furthermore, these approaches focus on finding emerging topics, requiring a chronological ordering of instances. In contrast, in our setting all documents are provided at once, before starting to detect subtopics interactively. Likewise, one-class classification, outlier detection and anomaly detection approaches are not applicable here [7], as their objective is the detection of abnormal, rare data points that differ from the data available at training time. Thus, they detect small rather than large groups, and do not aim to provide a meaningful clustering.

In topic detection and modelling, the aim is to discover the most relevant topics in text documents. An example is [4], where frequent topics in twitter messages are discovered. Another, more recent, is the discovery of the most interesting topics (and subtopics) in students' messages in a learning management system [16]. However, these approaches rely on temporal information. For example, the detected "emerging topics" in [4] and the "spike topics" in [16] correspond to terms have gained in frequency. More related is the detection of "chatter topics" in [16]. These chatter topics are sustained discussion topics, which are the most frequent words that are not in a set of so-called "DumpTerms". However, this requires providing a list of DumpTerms, which is unfeasible for large vocabularies.

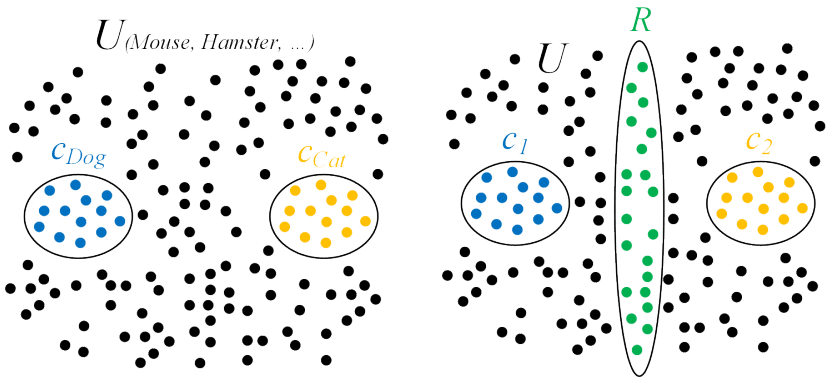

Figure 1: Dots represent single documents, some assigned to clusters of initially given subtopics. **(a) Left:** Example topic *animals*; c_{Dog}, c_{Cat} contain documents with subtopics *dog* and *cat*, respectively, remaining documents are unlabeled and might contain further animal-related subtopics. **(b) Right:** Building the residual set R of documents who's classification is uncertain.

3 Multi-Topic Clustering by Intent

We first explain the main ideas of our approach, before providing details. Similar to CBI [7], we use an uncertainty-related measure to construct a *residual set* of documents that are not certainly classified into one known subtopic. However, our aim is building a residual set consisting solely of documents from the intended topic. Thus, for constructing the residual set, we include a threshold on the document's *similarity* with respect to the union of known subtopics.

Having this residual set of documents, we extract the words that are used therein. In contrast to CBI, we exclude all words that have previously been rejected as subtopics. Furthermore, some terms are not specific for a subtopic, but are rather related to the topic as a whole. Therefore, we also exclude such words that occur frequently in all subtopics. We then compute a score for the remaining words. Similar to CBI, we consider the discriminatory power of a word, but we also consider the frequency of its co-occurrence with rejected words. The highest-ranking words in this combined score are proposed as new subtopics. Upon querying the users decision, the lists of rejected words and accepted subtopics are updated.

The pseudocode of our **M**ulti-Topic **C**lustering **b**y **I**ntent (MCBI) approach is given in Figure 3. Its input are lists of unlabelled U as well as labelled L documents. Using a bag-of-words representation over the vocabulary V, each document d therein has a feature vector $\vec{d}$. Furthermore, the current clustering C is given as a set of subtopics $s_1, s_2, \cdots$, each subtopic consisting of one or more words. Finally, a list of previously rejected subtopics V_{REJ} is given. These four lists are updated and returned by the algorithm.

The first step of the algorithm (lines 2–12) is the construction of the *residual set*. This is done by first training a Multinomial Naive Bayes Classifier (MNB) to classify documents into the different known subtopics (line 2). Iterating over each document u in the unlabelled set U (lines 3–12), we use this classifier to obtain a posterior-based score $p_{s|u}$, normalized by the vocabulary size, for each known subtopic s (line 6). For simplicity and speed, we consider in subsequent calculations solely the scores of the best (s) and second-best (s') subtopic (line 6). For better comparison with CBI, our algorithm uses the same uncertainty-based strategy to select the most ambiguous unlabelled documents. Thus, MCBI calculates for each unlabelled instance the difference between the score of its best s and second best s' subtopic (line 7)[1]:

$$uncertainty_u = p_{s|u} - p_{s'|u} \tag{1}$$

In a multi-topic corpus, one might distinguish three types of documents in the residual set: first, documents belonging to one of the *known subtopics*. Second, documents belonging to a *new subtopic* of the intended topic, and third, documents belonging to a *different topic*. Our aim is to identify documents of the second kind. For illustration, consider the example of the intended topic "sports", with known subtopics "soccer" and "tennis", the yet undiscovered subtopic "hockey", and the unrelated topic "building" (see Fig. 2 (a)). A classifier using the frequency of words like "soccer" or "tennis" might differentiate documents from the first type from the rest. However, it fails in distinguishing between the second and the third type, as both contain equally likely soccer- or tennis-specific words (see Fig. 2 (b)). Nevertheless, the second type of sports-related documents contain words that occur in both known subtopics, as for example the word "athlete". Such topic-specific words are more frequent in the second (and first) type, than they are in the third (see Fig. 2 (c)). Thus, the score over all subtopics will be greater for topic-related documents. We exploit this to remove topic-unrelated documents from the residual set by applying a threshold on the sum of scores (which we denote as *similarity*, line 8):

$$similarity_u = p_{s|u} + p_{s'|u} \tag{2}$$

Finally, all documents that are ambiguous with respect to being classified into one existing subtopic (high uncertainty) and are sufficiently similar to the union of the two subtopics in question (high similarity), are selected for the residual set R (lines 9–11 in Figure 3), i.e. R is a subset of the one computed with equation 1 by applying the similarity condition in order to guide pre-selected documents to the desired topic.

Having the residual set of topic-related documents, the next task is selecting candidate words for new subtopics, for which the user's feedback is then queried. Thus, we iterate over each word w occurring in a document of the residual set (lines 15–24). However, the most frequent words in the residual set might be specific for the topic, but not for a subtopic. Continuing the sports-topic above, an exemplary word is "athlete". Excluding such *topic-specific* words from being suggested as subtopic candidates saves annotation time and prevents them from being added to the list of rejected terms V_{REJ}. Such topic-specific words are frequent in the documents of the

[1] Here, entropy might be considered as another uncertainty measure.

labelled set L, too. Thus, we exclude any word that occurs in more than half of the labelled documents[2] (line 16). Here, the subfunction $getDocumentsWithWords(D, W)$ returns all documents in D that contain words W. Furthermore, our algorithm makes use of negative feedback in different ways. First, by excluding w in case it corresponds to a previously proposed but rejected subtopic (line 17). Second, by computing a score that considers how often this word co-occurs with rejected words. Thereby, we aim to exclude words that are specific to unrelated topics. For each rejected subtopic $n \in V_{REJ}$, we compute the co-occurence frequency of w and n and divide it with the frequency of n. Then, we use the maximum of this relative co-occurence frequencies as a *reject score* in line 20. Next, we compute a score reflecting the discriminatory power of the word w, similarly to [7]. This *discriminative score* corresponds to the difference between the number of unlabelled documents that are classified by the word w, subtracted by the number of labelled documents that are (wrongly) classified by w as discriminative feature (line 21). Finally, the best scoring among the candidate words are selected and added to the query list Q (line 25).

The user's feedback on these subtopic candidates is queried (line 27), which might result in an *accept* or *reject* decision. With the accepted words the list of subtopics and clustering is updated C (line 28), and their corresponding documents are moved from U to L (lines 29–30). Rejected words are simply added to the rejected word list V_{REJ} (line 31), and the resulting lists U,L,C,V_{REJ} are returned.

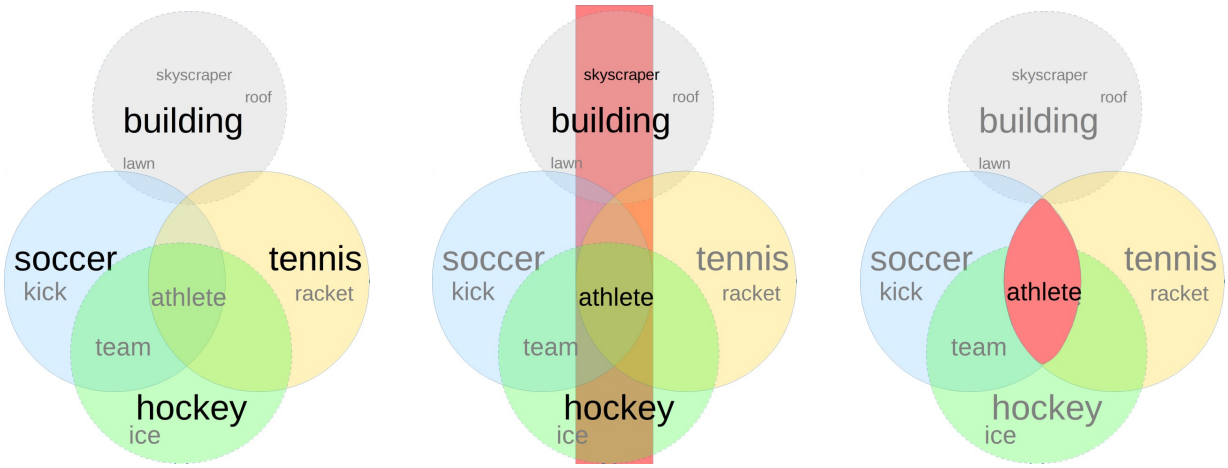

Figure 2: Illustration of subtopic's word bags. **(a) Left:** Overlap of word sets of the sport-subtopics "soccer" (given), "tennis" (given), "hockey" (unknown), with unrelated topic "building" (unknown). **(b) Center:** When considering solely ambiguity with respect to classification into "soccer" and "tennis", the residual set (in red) contains "building" and "hockey"-related documents. **(c) Right:** Considering also the similarity to the topic-common word "athlete", only "sport"-related documents like "hockey" overlap.

4 Evaluation

There are several questions that will be answered for one composed dataset to evaluate an active learning topic detection algorithm. How many subtopics will be found in comparison to random term selection? How many subtopics will be found over all subtopics in D? What is the round distribution, i.e. how many subtopics will be found over every round? Finally, how many unique subtopics will be found over rounds with different initial subtopics? An ideal algorithm would detect all subtopics that are available in D. In fact, it would detect them all in the first round without needing much time of a human oracle. Furthermore, it will indicate fast when it is not able anymore to find new subtopics and runs reliable, such that its performance is equal over differing starting values.

4.1 Dataset and Evaluation Design

For constructing a **dataset** with multiple topics, a list of 4309 common English nouns[3] has been parsed to search for Wikipedia articles which build up the document set D. In case of getting more than one search result on

[2]This threshold value $IGN = 0.5$ allows the algorithm to differentiate already with two documents in L, and also showed the best performance in experiments.

[3]Available at `http://www.desiquintans.com/nounlist`

function MCBI(U,L,C,V_{REJ})
  $MNB \leftarrow trainClassifier(L, C)$ ▷ train Multinomial Naive Bayes
  $R \leftarrow \emptyset$ ▷ **build residual set**
  for $u \in U$ **do**
    $p_{\cdot|u} \leftarrow posteriorScores(MNB, u)$ ▷ compute scores
    $(p_{best}, p_{2nd}) \leftarrow maxN(p_{\cdot|u}, 2)$ ▷ get the two highest scores
    $uncertainty_u \leftarrow p_{best} - p_{2nd}$ ▷ compute their difference
    $similarity_u \leftarrow p_{best} + p_{2nd}$ ▷ compute their sum
    if $(uncertainty_u \leq \tau_m) \wedge (similarity_u \geq \tau_s)$ **then**
      $R \leftarrow R \cup \{u\}$ ▷ add document to residual set
    end if
  end for

  $i \leftarrow 0$ ▷ **get words and their scores**
  for $w \in words(R)$ **do**
    if $|getDocumentsWithWords(L, w)| < 0.5 \cdot |L|$ **then** ▷ no topic term
      if $w \notin V_{REJ}$ **then** ▷ no rejected term
        $i \leftarrow i + 1$
        $word_i \leftarrow w$ ▷ compute scores
        $rejscore_i \leftarrow \max_{n \in V_{REJ}} \left(\frac{|getDocumentsWithWords(U \cup L, \{w,n\})|^2}{|getDocumentsWithWords(U \cup L, n)|} \right)$
        $discscore_i \leftarrow |getDocumentsWithWords(U, w)| - |getDocumentsWithWords(L, w)|$
      end if
    end if
  end for
  $Q \leftarrow getBestScores(word, discscore, rejscore)$ ▷ select candidates

  $(Q_{accepted}, Q_{rejected}) \leftarrow queryUser(Q)$ ▷ **query user's feedback**
  $C \leftarrow C \cup Q_{accepted}$ ▷ update clustering
  $L \leftarrow L \cup getDocumentsWithWords(U, Q_{accepted})$ ▷ update labelled set
  $U \leftarrow U \setminus getDocumentsWithWords(U, Q_{accepted})$ ▷ update unlabelled set
  $V_{REJ} \leftarrow V_{REJ} \cup Q_{rejected}$ ▷ update reject list
return U,L,C,V_{REJ}
end function

Figure 3: The MCBI Algorithm

Wikipedia, the first choice has been considered. Stop words have been removed, followed by lemmatization. On D, *tf-idf* with a threshold of 5 words per document was applied to keep solely the most important words. The resulting vocabulary set V has a size of 6.594 unique terms, which were considered in the bag-of-words representation. In order to enable a fair and automatic evaluation, a topic with known subtopics as ground truth was required. We have chosen the topic *countries*, as it is closed and identifiable. This means, that a complete list of all countries' names (as subtopics) is obtainable. We also wanted to count in *languages* [4] and *denonyms* (naming of a country's natives) [5] because of their high relatedness. As in most natural language applications some words are ambiguous, e.g. the country *Georgia* which is also an US state. For simplicity, such ambiguous terms were considered as valid subtopics. Furthermore, countries like *Marshall Islands* and *New Zealand* were transformed into single terms like *Marshall* and *Zealand* to reduce ambiguity.

We compared our MCBI approach (denoted as *MCBI, NF=on*) against three **baselines**: *CBI* [7][6], which in the paper [7] was already shown to compare favourably against other clustering-based approaches. *Random* selection, and finally a variant of MCBI without using negative feedback (denoted as *MCBI, NF=off*). For CBI's residual set size $|R|$, different parameter values were used (see first column in CBI's results table 1a). On this dataset, we run *CBI* and the *MCBI* algorithms 20 times, while *random* was run 1000 times to get reliable results.

[4] Available at `http://www.infoplease.com/ipa/A0855611.html`
[5] Available at `http://geography.about.com/library/weekly/aa030900a.htm`
[6] We reimplemented this approach, as the original source code is not available.

Each of those iterations consisted of 20 sampling rounds, where in each round 20 subtopics where queried. For the experiments we used 2, respectively 4, initial given subtopics. Within iterations, initial subtopics changed between countries (e.g. *Germany* & *Portugal*). When constructing *MCBI*'s residual set R, we iteratively lowered the thresholds τ_s and τ_m until the residual set's size was equal or greater than one percent of the unlabelled set's size. We also evaluated the effect of choosing not 0.5 as threshold IGN for ignored topic terms in line 16, which confirmed the our choice.

4.2 Results and Discussion

We first provide the aggregated results in Fig. 4, where we compare the four algorithms in terms of their average (over all iterations) number of found subtopics (ordinate axis) in each round (abscissa). In these experiments, *Random* yields a nearly uniform performance over the rounds. This is as expected, as due to the large size of V there is little dependence between rounds (i.e. it corresponds to sampling without replacement from a very large urn). *Random* performs on this dataset consistently better than *CBI*. This is in contrast to single-topic environments where CBI originates from. This evaluation shows the necessity of adaptations for handling multi-topic environments. Furthermore, *Random* performs worse than *MCBI* in most rounds. The performance of *MCBI* increases over the first rounds, before starting to decline towards the last quarter of the rounds. In particular the *MCBI* variant using negative feedback improves at the beginning. This indicates that the negative feedback helps in determining the relevance of subtopics. However, in the last quarter both *MCBI* variants perform similarly. This indicates that at this point negative feedback might start to be too restricting, although this requires further investigation.

When relating the maximum average found subtopics 26.20 to the total number of available subtopics (183) in D, ca. 14% are discovered, with standard deviation 4.69 over all iterations. Another interesting information is the unique count of subtopics in relation to the total availability over all iterations within highest scoring settings ($\#subtopics = 2, 4; NF = on; IGN\ Share\ variable$) which ranges between 60 and 66. Since we already detect up to 26.20 subtopics in one iteration, it seems not worth to restart with differing initial values depending on application.

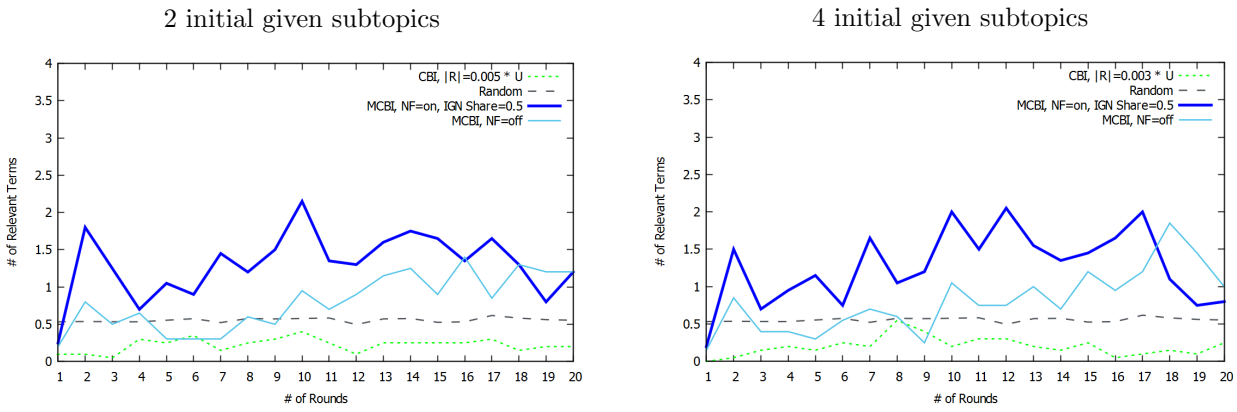

Figure 4: Average number of found subtopics over 20 rounds with original CBI, 2 & 4 initial subtopics and varying size of R

More detailed results for *CBI* and random are given in Table 1a. In the first column, settings are listed to compare different numbers of initial given subtopics and the document size of R. In the other columns, the average number of found countries, languages & denonyms and the count of subtopic and related words over all iterations with different initial given subtopics are depicted. Selecting words at random gives much better results than applying CBI in its original form. When composing R, documents are rated best for which it is uncertain how to associate them to the given subtopic clusters. Since we work in a multitopic environment, R will mostly imply documents that do not share any words with those from L, posteriors for given clusters and $u \in U$ are nearly equal which will result in small margins. For new subtopic proposals, those words are preferred that occur most often in R and least often in L. Since R is multitopical, most frequent words from R with a high unrelatedness to the searched topic are proposed. The size of R is set very low to limit the number of unrelated documents and therefore to achieve best results under these critical circumstances. The more documents we consider for R, the worse gets performance neglecting variance.

Detailed results for *MCBI* on different configurations (number of initial subtopics, negative feedback and ignoring shares) are given in Table 1b. *MCBI* without using negative feedback finds 44% more subtopics than *random*. By incorporating negative feedback without factoring in ignored words, we again make a jump with a total improvement of 117%. Furthermore, the choice of setting the IGN threshold to 0.5 yields the best results, as explained in Section 3.

Iteration Setting # subtopics, $\|R\|$	Average subtopics found		
	Count.	Lang. & Denonym	Total
Random	5.52	5.60	11.12
$2, 0.003 \cdot \|U\|$	1.2	3.25	4.45
$2, 0.005 \cdot \|U\|$	0.9	3.9	4.8
$2, 0.01 \cdot \|U\|$	0.4	3.15	3.55
$2, 0.05 \cdot \|U\|$	0.05	2.4	2.45
$2, 0.1 \cdot \|U\|$	0.0	2.3	2.3
$4, 0.003 \cdot \|U\|$	1.05	2.95	4.0
$4, 0.005 \cdot \|U\|$	0.9	2.8	3.7
$4, 0.01 \cdot \|U\|$	0.95	2.85	3.8
$4, 0.05 \cdot \|U\|$	0.0	1.45	1.45
$4, 0.1 \cdot \|U\|$	0.1	2.15	2.25

(a) Detailed evaluation results for CBI and random.

Iteration Setting # subtopics, NF, IGN	Average subtopics found		
	Count.	Lang. & Denonym	Total
Random	5.52	5.60	11.12
$2, off, -$	7.75	8.20	15.95
$2, on, 0$	15.40	8.60	24.00
$2, on, 0.1$	15.55	8.9	24.45
$2, on, 0.3$	15.35	8.9	24.25
$2, on, 0.5$	15.90	10.30	26.20
$2, on, 0.7$	15.60	10.15	25.75
$2, on, 1.0$	10.15	7.05	17.20
$4, off, -$	9.00	7.10	16.10
$4, on, 0$	15.40	7.90	23.30
$4, on, 0.1$	15.90	8.00	23.90
$4, on, 0.3$	15.95	8.45	24.40
$4, on, 0.5$	16.30	9.05	25.35
$4, on, 0.7$	15.90	9.35	25.25
$4, on, 1.0$	9.00	6.90	15.90

(b) Detailed evaluation results for $MCBI$ and random.

Table 1: Detailed results

5 Conclusion

This work addressed the clustering by intent scenario recently introduced in [7]. For this scenario, an active approach for detecting subtopics was presented. This approach extends the CBI algorithm to multi-topic environments, and to the incorporation of positive and *negative* user feedback. It iteratively constructs so-called residual sets of documents within the same topic. Based on this residual sets, it derives candidates for new subtopics. Then, feedback is actively queried from the user on these candidates. Finally, this is used to refine the subtopic and residual sets in the next iteration. The approach was evaluated on a corpus of Wikipedia articles for the 4309 most common English nouns, comprising a broad range of different topics. In our experiments, MCBI provides an improvement of up to 137% against random sampling, and 445% against CBI. While these first results are promising, a more extensive experimental evaluation is planned for the future. Because of the unsupervised learning task, a way to automatically tune parameters like uncertainty and similarity for constructing the residual set as well as IGN with differing datasets is desired. Furthermore, the role of negative feedback in such a system seems worth to be investigated further.

Acknowledgements

We would like to thank Andreas Nürnberger, Daniel Kottke, and George Forman for insightful discussions on this topic.

References

[1] An active learning framework for semi-supervised document clustering with language modeling. *Data and Knowledge Engineering*, 68(1):49–67, 2009.

[2] Martin Atzmüller. Subgroup discovery. *WIREs Data Mining Knowledge Discovery*, 5(1):35–49, 2015.

[3] Mohamed-Rafik Bouguelia, Yolande Belaïd, and Abdel Belaïd. Efficient active novel class detection for data stream classification. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, number 3, pages 2826–2831, 2014.

[4] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporaland social terms. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, pages 4:1–4:10, 2010.

[5] Derya Dinler and Mustafa Kemal Tural. *A Survey of Constrained Clustering*, pages 207–235. Springer International Publishing, Cham, 2016.

[6] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert Nowak. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 260–268, 2011.

[7] George Forman, Hila Nachlieli, and Renato Keshet. Clustering by intent: A semi-supervised method to discover relevant clusters incrementally. In Albert Bifet, Michael May, Bianca Zadrozny, Ricard Gavalda, Dino Pedreschi, Francesco Bonchi, Jaime Cardoso, and Myra Spiliopoulou, editors, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part III*, pages 20–36, Cham, 2015. Springer International Publishing.

[8] Dragan Gamberger, Nada Lavra, and Goran Krstai. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, 28:27–57, 2003.

[9] Thomas Hofmann and Joachim M. Buhmann. Active data clustering. In *Proceedings of the 10th Conference on Advances in Neural Information Processing Systems*, NIPS '97, pages 528–534. MIT Press, 1998.

[10] Janez Kranjc, Jasmina Smailović, Vid Podpečan, Martin Grčar, Miha Žnidaršič, and Nada Lavrač. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing and Management*, 51:187–203, 2014.

[11] Akshay Krishnamurthy. *Interactive Algorithms for Unsupervised Machine Learning*. PhD thesis, 2015.

[12] Mohammad Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani M. Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Trans. on Knowl. and Data Eng.*, 23(6):859–874, June 2011.

[13] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham. Classification and novel class detection in data streams with active mining. In *Proceedings of the 14^{th} Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, PAKDD 2010, pages 311–324, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

[15] Burr Settles. *Active Learning*. Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[16] Llanos Tobarra, Antonio Robles-Gmez, Salvador Ros, Roberto Hernndez, and Agustn C. Caminero. Discovery of interest topics in web-based educational communities. In *Proceedings of the International Symposium on Computers in Education (SIIE)*, pages 87–92. IEEE, 2012.

[17] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proc. of the 1st Europ. Symposium on Principles of Data Mining and Knowledge Discovery*, pages 78–87. Springer, 1997.

[18] Xiatian Zhu, Chen Change Loy, and Shaogang Gong. Constrained clustering with imperfect oracles. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1345–1357, 2015.

1: **function** CBI(U,L,C)
2: $MNB \leftarrow trainClassifier(L, C)$ ▷ train Multinomial Naive Bayes
3: $R \leftarrow \emptyset$ ▷ **build residual set**
4: **for** $u \in U$ **do**
5: $p_{\cdot|u} \leftarrow posteriorScores(MNB, u)$ ▷ compute scores
6: $(p_{best}, p_{2nd}) \leftarrow maxN(p_{\cdot|u}, 2)$ ▷ get the two highest scores
7: $uncertainty_u \leftarrow p_{best} - p_{2nd}$ ▷ compute their difference
8: **if** $(uncertainty_u \leq \tau_m)$ **then**
9: $R \leftarrow R \cup \{u\}$ ▷ add document to residual set
10: **end if**
11: **end for**
12:
13: $i \leftarrow 0$ ▷ **get words and their scores**
14: **for** $w \in words(R)$ **do**
15: $i \leftarrow i + 1$
16: $word_i \leftarrow w$ ▷ compute scores
17: $discscore_i \leftarrow |getDocumentsWithWords(U, w)| - |getDocumentsWithWords(L, w)|$
18: **end for**
19: $Q \leftarrow getBestScores(word, discscore)$ ▷ select candidates
20:
21: $(Q_{accepted}) \leftarrow queryUser(Q)$ ▷ **query user's feedback**
22: $C \leftarrow C \cup Q_{accepted}$ ▷ update clustering
23: $L \leftarrow L \cup getDocumentsWithWords(U, Q_{accepted})$ ▷ update labelled set
24: $U \leftarrow U \setminus getDocumentsWithWords(U, Q_{accepted})$ ▷ update unlabelled set
25: **return** U,L,C
26: **end function**

Figure 5: The Implemented CBI Algorithm (based on [7]).