Using Clustering for Categorization of Support Tickets

Daniel Beneker and Carsten Gips

FH Bielefeld - Campus Minden,
Artilleriestraße 9, 32427 Minden, Germany
http://www.fh-bielefeld.de/

Abstract. Support tickets from customers contain much hidden information. Unsupervised machine learning methods are able to discover this hidden information. In this paper we propose the categorization of support tickets using clustering methods in combination with topic models. Furthermore label generation techniques are used to generate meaningful names for these categories. The results are compared with related research.

Keywords: clustering, support tickets, categorization, topic model, nmf, k-means, information retrieval

1 Introduction

The volume of digital data generated annually has grown massively in recent years, but only a small amount of this data is used for analysis. Manual analysis of these data sets is not only more difficult, but also more cost-intensive. Therefore, it is important to organize and categorize them appropriately. These tasks can be successfully automated with clustering algorithms.

At *Mittwald CM Service GmbH & Co. KG* a large part of the communication with the customer takes place over the ticket system of the company. The customer makes a new request via the ticket system, the so-called support ticket. This is handled by employees, so that usually a conversation sequence consisting of several internal and external tickets is created. External tickets are tickets from the customer as well as responses of the employees to the customer. Internal tickets contain additional information and are only visible to employees.

Since these tickets are currently not analyzed, it is difficult to evaluate for which topic the company receives the most requests. This paper presents an approach to cluster support tickets into categories. The algorithms k-means and NMF are used for this purpose, which belong to the field of unsupervised learning and do not require labelled training data. In a second step label generation techniques are used to generate meaningful names for the calculated categories. These meaningful names are important because, without them, it is not possible to evaluate which topics cost how much time or how much money. With the help of the named categories, it is also possible that an employee, who is an expert in a particular topic, processes the corresponding tickets prioritized.

CEUR-WS.org/Vol-1917/paper10.pdf

Copyright © 2017 by the paper's authors. Copying permitted only for private and academic purposes.
In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
Rostock, Germany, 11.-13. September 2017, published at http://ceur-ws.org

In Section 2 we briefly discuss different approaches for clustering documents and present in section 3 our approach to cluster the support tickets and to extract meaningful names for the found clusters. The results are evaluated in section 4.

2 Related Work

There are many different approaches to categorize texts and documents.

In [1] Blei uses the LDA (Latent Dirichlet Allocation) algorithm to categorize 17.000 articles from the journal *Science* into 100 categories. The found categories provided a capable and general overview of the individual subjects. LDA belongs to the topic models as well as the algorithm NMF used in this work. An extension to LDA is described in [9]. The hierarchical variant presented there was used to categorize news texts around the disappearance of the Malaysia airline flight *mh-370*. As resulting categories the individual theories about the status of the plane like *plane crash* or *terrorism* were obtained.

In [4], Kuang et al. show how NMF (Nonnegative Matrix Factorization) can used to classify documents. They also test NMF on five different data sets like the well-known sets *20Newsgroups* and *Reuters* and compare the results to a baseline k-means clustering. In four out of five cases, NMF yielded better results. In [8] Shahnaz et al. note, that the accuracy of the clustering strongly depends on the number of clusters and the dataset. With only 2 categories of the *Reuters* dataset, an accuracy of 99% was achieved. With 20 categories it is only 54%. But on another dataset, they received over 80% accuracy with 20 categories.

Other approaches attempt to generate training data in a first step using cluster algorithms and keyword-lists, and to train a supervised classifier such as SVM or Naive Bayes in a second step. The classifier is then used to categorize documents. So they combine unsupervised learning methods with methods of supervised learning. In [2] Ko and Seo split the documents into sentences and categorize them by a list of keywords. Those sentences are used to train a naive bayes classifier. The results did not differ significantly from those of pure supervised learning. But the approach is helpful when training data must be generated. An approach that uses automatic created keywords to create training data is described in [3]. They evaluate their approach on the *20Newsgroups* and *Reuters* dataset and achieved the best results using only a small number of keyword. Furthermore, they received a small improvement, if humans filter those automatic created keywords.

The algorithm DCF (Description Comes First) is described in [10]. That approach is different from the others. First, label candidates are generated for each text. Then the texts are categorized (for example with k-means). The final step the centroids are used to select one of the label candidates.

3 Approach

In order to categorize support tickets, the tickets are preprocessed and converted into feature vectors, and cluster algorithms are applied for categorization. In the

last step meaningful names are assigned to the categories (see Figure 1). After initial clustering, the trained algorithm can be saved to get fixed categories and to predict the category of new tickets later.

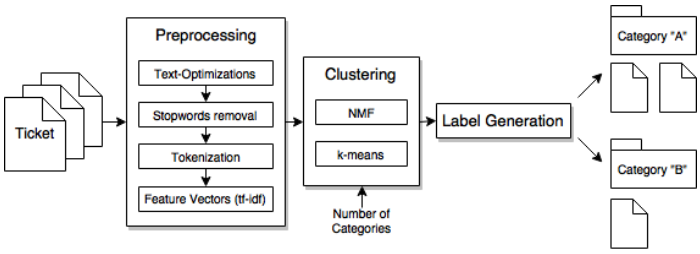


Fig. 1: Approach Overview

3.1 Preprocessing

This section describes how the support tickets are preprocessed so that they can be used by algorithms such as k-means and NMF.

Normalization A support ticket consists of two parts, the subject and the message. For further processing subject and message are concatenated and converted to lower case. Support tickets can contain many elements that can adversely affect the quality of clustering. After loading the tickets, elements like URLs, e-mails and account or database names are removed using regular expressions. Single numbers are also removed, as we want strings as categories.

Stopword removal Stopwords contain little or no information and can therefore be removed. We used the stopword list from the NLTK framework[1], extended by a list containing words which could have a negative impact on the automatic naming of the clusters, such as salutation and greeting formulas. Especially short tickets could otherwise be assigned to a common cluster called *freundliche Grüße*.

Text tokenization To split the normalized texts into token lists we used a simple tokenizer with the following regular expression: `/\w\w\w+/u`, to capture all words of length 3 and longer. Other characters such as punctuation are discarded, since the cluster labels should not contain punctuation marks.

[1] `http://www.nltk.org/`

Generation of feature vectors To cluster texts using algorithms, they must first be converted into numbers. One ways is to use the *term frequency - inverse document frequency (tf-idf)* vectors. These feature vectors are based on two assumptions. First, words that are more common in a document are more important, or they describe the document better, and second, words that are common in all documents are unimportant. The tf-idf value of a word t in a document d is calculated as follows:

$$\text{tf-idf}_{t_d} = tf_{t,d} \times idf_t \tag{1}$$

$tf_{t,d}$ specifies the occurrences of a word t in a document d. This value is then weighted with the term idf_t. This is calculated as follows:

$$idf_t = log(\frac{N}{df_t} + 1) \tag{2}$$

N is the number of all documents. df_t stands for *document frequency* and specifies in how many documents the word t occurs. For rarely used words, the idf_t value is large; for common words the value is small.

After this step, each document is represented by a feature vector with the length of the vocabulary. At this point, if desired, all words above and below a certain occurrence frequency can be specifically excluded in order to reduce the length of the feature vectors. This has the advantage that, for example, words with spelling mistakes, which occur very rarely, are filtered out.

3.2 Algorithms

We use k-means because it is a simple baseline algorithm and NMF because it belongs to the topic models and provides descriptive results as we want to compare a standard cluster algorithm and a topic model.

k-means K-means represents a simple algorithm to classify data into clusters. It can be divided into 3 steps. In the first step, the cluster centroids $\{c_1, \cdots, c_K\}$ are initialized for randomly. In the second step, each data point x is assigned to the cluster with the smallest distance. Following, the cluster centroids are recalculated in a third step. The centroid is always calculated from the average of all the data points assigned to it. Steps 2 and 3 are repeated until cluster centroids no longer change.

Non-negative Matrix Factorization (NMF) Non-negative matrix factorization describes a method in which a matrix is represented by a product of two smaller matrices. This method can also be used to cluster documents. [8]

Given a matrix V, it can be decomposed with NMF into the matrices W and H, so that $W \times H \approx V$. If V is the size $m \times n$, then W has the size $m \times k$ and H

the size $k \times n$. The goal is that the product of W and H should match the original matrix V. From this the following minimization problem can be derived: [8]

$$\min_{W,H} \|V - WH\|_F^2 \tag{3}$$

This minimization problem can be solved by a special gradient descent method as described by [6]. $\|\cdot\|_F$ stands for the Frobenius norm. It specifies the size of a matrix and is defined as:

$$\|K\|_F = \sqrt{\sum_i \sum_j |k_{ij}|^2} \tag{4}$$

This method can also be used for clusters of documents. Figure 2 shows what information the matrices V, W, and H contain. The matrix V contains all the documents to be used for clustering. A document is always represented by a number vector of length n. n corresponds to the length of the vocabulary over all documents in V. As a rule, vectors with weighted word frequencies, such as *tf-idf*, are used. m stands for the number of documents. An entry V_{mn} from V thus contains a value that reflects the meaning of the word n in the document m (if *tf-idf* is used). K in the smaller matrices W and H stands for the number of clusters. As with k-means, this value must be set by the user beforehand. After decomposing V into the matrices W and H, W contains the assignments of the individual documents to the clusters. The largest number in each line decides about the belonging to a cluster. H contains a probability distribution of words over clusters. The larger the entry H_{ig} in H, the more likely it is that the word g occurs in the documents from cluster i. The matrix H can be used to generate names or descriptions for the respective clusters. [8]

NMF can be assigned to the topic models rather than to the classical cluster algorithms, such as k-means. Topic Model, similar to cluster algorithms, can find patterns in documents. They do not split the documents into disjoint clusters, but into so-called topics, that is, Clusters that overlap. For example, a document can belong to 90% of Topic 1, to 8% of Topic 3, and 2% of Topic 2. The subdivision into disjoint clusters with a topic model can be done by assigning a document only to the topic with the highest value. [1]

3.3 Label generation

Without descriptive cluster names, it is difficult for humans to gather information from the grouped documents. One of the most difficult tasks in document clustering is therefore to generate a description which is understandable to humans. For algorithms from the field of supervised learning this is easily possible via the training data. However, these are not present during clustering.

The naming of clusters is often referred to as *cluster labeling* in the literature and can be done in many different ways. They can be divided into manual, automatic and semi-automatic approaches. The manual naming of clusters is done by the human being who creates a name or description after visual inspection of

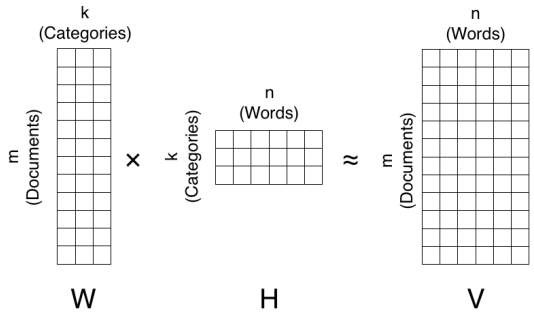


Fig. 2: Graphical representation of NMF

some documents of a cluster. Automatic methods attempt to generate descriptions from the most powerful features within a cluster. [5] [10] Semi-automatic approaches combine these two possibilities, for example, by customizing the automatic descriptions by the human being. Another possibility is that a human describe categories with lists of keywords. Subsequently, the documents are automatically sorted into these categories. [2, 3] Other approaches use additional external information, such as the WordNet database, which provides lexical-semantic relationships between words. [11]

The implemented solution within the scope of this work consists of an automatic generation of labels and offers the option to adjust the labels manually afterwards. Figure 3 introduces the individual steps, which are described in more detail below.

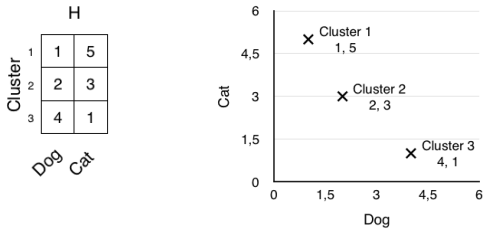
Fig. 3: Cluster-labeling-steps

Extraction of label candidates The first step of cluster labeling is to extract label candidates. For each cluster, all words in the vocabulary are sorted according their relevance to the cluster. When using NMF to cluster the documents, the H matrix already contains all the information you need. For each word and cluster, a H_{ig} in H entry contains a value that reflects the relevance of the word g in cluster i (see section 3.2). It is therefore sufficient to sort the matrix H line by line. The larger a value, the more important is the corresponding word.

For k-means as a clustering algorithm, this information can be obtained via the cluster centroids. The cluster centroids are located in the feature space and therefore have the same dimension as the tf-idf feature vectors. In order to obtain

a list of words sorted by relevance for k-means, the values of the cluster centroids must be sorted.

Figure 4 shows the matrix H with three different feature vectors. They each contain the two features *dog* and *cat*. For example, to obtain all label candidates for cluster 1, the values of the row are sorted descending. On the right, the image shows the characteristic space with the found cluster centroids. A row of the matrix H can be compared to a cluster center. Sorting the values of a cluster center thus corresponds to sorting a row in matrix H.



(a) NMF (b) k-means

Fig. 4: Graphical representation of the feature vectors / cluster centroids

POS-Tagging Through POS tagging (Part-Of-Speech-Tagging), it is possible to filter automatically according to certain types of words. When naming clusters, this is very handy, as it can effectively filter out unwanted words. The Stanford POS tagger[2] was used for this work, which in addition to English also supports the German language by using the Stuttgart-Tübingen Tag Set (STTS) [7]. The Stuttgart-Tübingen tag set contains over 50 different tags and can, for example, also distinguish between different types of conjunctions.

The following tags were used:

NN Normal noun, such as table, apple, travel
NE Proper noun, such as Smith, Hamburg
FM Foreign language material

Support tickets contain many technical terms, which often come from English, such as *domain record* or *virtual host*. In order to exclude these terms as a description for a cluster, in addition to nouns and their own names, they were also filtered according to foreign-language material. All words that do not belong to any of the tags are excluded. For example, POS tagging would remove the words *erreichbar* and *seit* from the list *server*, *erreichbar*, *fehler*, *managed*, *seite*, *problem*, *ftp*, *seit*, *ssh*, *datenbank*.

[2] `https://nlp.stanford.edu/software/tagger.shtml`

Stemming We used the Snowball Stemmer[3] for stemform reduction. For words whose word stem occurs more than once, only the original of the first occurrence was retained. The list *domain*, *domains*, *relocation* becomes after stemming *domain*, *relocation*.

Manual adjustment Manual adjustment is the last step in cluster naming. After the word lists have been filtered with the aid of POS tagging and stemming, the most meaningful word, as well as a list of the n most meaningful words in a file, is stored for each cluster. The entries can be customized by the user.

4 Evaluation

This chapter contains the results of categorization using k-means and NMF. The data set contains only customer requests. This means that only the first message, which the customer writes, is contained. Tickets from employees, as well as automatically generated tickets, are not included. This data set was selected because the categorization of the customer requests was the most useful. For example, if new customer requests are categorized directly, they can be edited by employees who have the most expertise in the category. To include all the tickets form conversation would be counterproductive at this point, since only the first ticket of the customer is available at the time of the categorization of new tickets. The data set contains the latest 50,000 tickets with an average length of 680 characters. The tickets thus cover a period of approximately one year.

Figure 5 shows the results of categorization with k-means and NMF in 30 categories. The categorization was also executed with 50 and 70 categories. Because of the size, the results are not included.

Three different aspects are analyzed for the evaluation, the intersection of categories at k-means and NMF, the distribution of tickets per category, and the development of a category when the number of clusters changes.

4.1 Intersection of categories at k-means and NMF

Both algorithms, k-means and NMF, find similar categories and there is a large intersection between the categories. In categorization with 30 categories, 21 categories are found in the results of both algorithms. In categorization with 50 categories, 33 are equivalent. With 70 categories, there are still 46 of categories equivalent. By increasing the number of categories from 30 to 50, the number of identical categories is decreasing in percentage. When increasing from 50 to 70 categories, there is no longer a large percentage change. With 30 categories, there are 70%, with 50 categories 66%, and with 70 categories 65.7%.

For the categorization of the tickets in 30 categories, the intersection of tickets was examined. For all categories found by both k- means and NMF, it was

[3] `http://snowball.tartarus.org/`

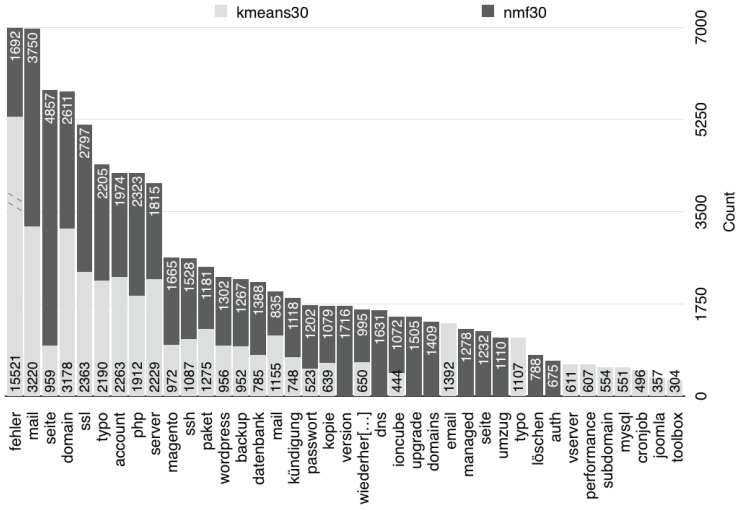


Fig. 5: Number of tickets per category - k-means vs. NMF (30 categories)

checked how many tickets are included in both categories (see fig 5). Categories that are duplicated under the same or a similar name have been summarized for the calculation of the intersection. For some categories, such as *ssl*, the intersection is relatively high. Only about 100 tickets from over 2000, which fall into category *ssl* at k-means, are not in category *ssl* at NMF. In some categories, however, the number of tickets included varies greatly. In the category *password*, NMF has more than twice as many tickets as with k-means. However, the category at NMF contains almost all the tickets, which also fall into this category at k-means. With k-means, some of the small categories are even smaller than with NMF.

4.2 Distribution of tickets per category

To better evaluate the distribution of tickets per category, the number of tickets per category is shown in Figure 5. The diagram shows the distribution from the categorization in 30 categories. It is striking that there is a category with a lot of tickets at k-means. When categorized into 30 categories, the largest category contains almost 16.000 tickets. With NMF, the largest category, however, contains only about 5000 tickets, that is only a third of the tickets. A similar pattern can be observed at the categorization with 50 and 70. In 70 categories, the largest category, with nearly 10,000 tickets, contains a fifth of the complete data set. For NMF, however, the largest category is only about 1500 tickets. The difference between k-means and NMF in the largest categories also increases with increased cluster number. Also it is interesting that the name of the largest category in k-means varies depending on the number of clusters. The names of the largest categories are *fehler*, *seite*, and *account*. NMF tends to find categories that are

similar in size. There are no very large categories and no very small categories. In order to better evaluate whether k-means further reflects the actual structure of the tickets, the large category *fehler* was further analyzed. All tickets assigned by k-means to the category *fehler* have been categorized with NMF. Only 7% of the tickets were assigned with NMF in the same category *fehler*. The largest block with approximately 27% of tickets falls with NMF into the category *seite*. The other tickets split relatively evenly in 26 other categories. One possible theory for the large categories is that with k-means *outliners*, i.e. tickets, which are located at the edge of a cluster, are grouped.

4.3 Development of a category when the number of clusters changes

So far differences between the two algorithms k-means and NMF have been analyzed. In this section we will examine the development of a category at different clusters, but using the same algorithm. The *domain* and *domains* categories were selected as examples. The term *domain* is very general in itself and is experienced in many other areas such as e-mail, SSL certificates or DNS. For this reason the category *domain* is particularly interesting. All 50,000 tickets of the data set were divided into 30, 50 and 70 categories for evaluation with NMF. In 30 categories, a total of 4020 tickets fall into the category *domain(s)*. Out of 4020 tickets, 78% of tickets are again in the *domain(s)* category. The remaining tickets are divided into 22 other categories. The right column shows the categorization with 70 categories. Only 38% of the tickets fall into the category *domain(s)*. Overall, the tickets are now divided into 39 categories. Except for the category *ftp*, all categories from nmf50 were also found by nmf70. That means a larger amount of tickets does not suddenly change into another category, which would be an argument against the quality of the categorization.

5 Advantage of this solution

Compared to a keyword driven approach, categorization with clustering algorithms has a great advantage: Tickets, which do not contain the name of a category, can be assigned to a category. Tickets of this kind are difficult to categorize through simple procedures such as a keyword search, where many/some keywords from predefined lists of keywords for a category need to be present in a text to label it. Ko and Seo used in [3] a keyword driven approach, but would also able to assign a ticket to a category, although the name of the category is not present in the ticket, because they only created training data using the keywords. The difference is that they first had to create categories or keywords, which is not necessary in this approach.

Listing 1.1 shows a ticket which has been assigned by NMF to the category *typo* (as a result of categorization in 30 categories). *typo* stands for the content management system TYPO3, which is used by many Mittwald customers.

Listing 1.1: Example Ticket - Category *typo*

```
Pfad Sendmail
Hallo! Wie lautet der Pfad zu Sendmail?
Beste Grüße xxxxxx xxx
```

The ticket itself does not contain the word *typo* and also no obvious keyword, which suggests a relation to TYPO3. However, *Sendmail* is a software to send e-mails, which is often used in TYPO3 for this purpose. The path to *Sendmail* must be entered in TYPO3. Our approach allows this ticket still to be assigned automatically to a meaningful category with the help of algorithms like k-means or NMF.

Other examples are shown in Listings 1.2 and 1.3, where a mail-related and a ssl-related ticket were correctly recognized and assigned to the category *mail* and *ssl*, resp.

Listing 1.2: Example Ticket - Category: *mail*

```
Required TLS im Required Mode
Sehr geehrtes Mittwald Support Team
ein Kunde wollte wissen, ob die Mittwald Postfächer als Verschlüsselungsart
"TLS im Required Mode" unterstützen. Ich weiss nicht ob es die Starttls
Methode ist.
Vielen Dank und Gruss xxxxxx xxxxxxx
```

Listing 1.3: Example Ticket - Category: *ssl*

```
Let's encrypt
Wie kann ich bei Mittwald eine Let's encrypt Verschlüsselung nutzen?
```

The ticket in Listing 1.4 has been assigned to the category *php*. In this case a keyword driven approach would have failed.

Listing 1.4: Example Ticket - Category: *php*

```
Execution Time
Hallo! Bitte ändern:
max_execution_time:     30 Recommended min value 120
max_input_vars: 1000     Recommended min value 3000
Danke lg xxxxxxx
```

Assuming manual categorization takes about 4 seconds and there are about 120 customer requests per day, then 8 minutes of time can be saved per day. The automatic categorizing of inventory tickets saves approximately 110 hours.

It should also be noted that the categories found in this domain are quite general and thus give a good overview of the topics that the customers deals with.

6 Conclusion and Future Work

The goal of this work was the unsupervised categorization of support tickets. According to latest research, there is no distinct solution for this. Especially the automatic naming of the calculated categories is not easy to solve. The result of a categorization is strongly dependent on the data set and the cluster algorithm. In addition, the quality of the categorization is difficult to evaluate. This work

has shown that a fully automatic categorization, including the naming of the categories, is feasible and the found categories reflect meaningful subjects.

To improve the quality of the found categories we propose hierarchical clustering of the support tickets. Such a hierarchy could contain more information than a flat clustering. Each category can be divided into other (sub-) categories, providing a more detailed view of the data.

Acknowledgments

We would like to thank *Mittwald CM Service GmbH & Co. KG* for the cooperation and the provision of the data set.

References

1. Blei, D.: Probabilistic Topic Models. Communications of the ACM 55(4), 77–84 (2012)
2. Ko, Y., Seo, J.: Automatic Text Categorization by Unsupervised Learning. In: Proceedings of the 18th Conference on Computational Linguistics. vol. 1, pp. 453–459. Association for Computational Linguistics (2000)
3. Ko, Y., Seo, J.: Learning with Unlabeled Data for Text Categorization Using a Bootstrapping and a Feature Projection Technique. In: Proc. 42nd Meeting of the Association for Computational Linguistics (ACL'04). pp. 255–262. Association for Computational Linguistics (2004)
4. Kuang, D., Choo, J., Park, H.: Nonnegative matrix factorization for interactive topic modeling and document clustering. In: Partitional Clustering Algorithms, pp. 215–243. Springer International Publishing (2015)
5. Li, X., Chen, J., Zaiane, O.: Text document topical recursive clustering and automatic labeling of a hierarchy of document clusters. In: Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD. pp. 197–208 (2013)
6. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. 19(10), 2756–2779 (Oct 2007)
7. Schiller, A., Teufel, S., Stöckert, C.: Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset) (1999), `http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf`
8. Shahnaz, F., Berry, M.W., Pauca, V.P., Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Information Processing & Management 42(2), 373–386 (Mar 2006)
9. Smith, A., Hawes, T., Myers, M.: Hiérarchie: Interactive Visualization for Hierarchical Topic Models. In: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. pp. 71–78. Association for Computational Linguistics (2014)
10. Stefanowski, J., Weiss, D.: Comprehensible and accurate cluster labels in text clustering. In: Large Scale Semantic Access to Content (Text, Image, Video, and Sound). pp. 198–209 (2007)
11. Tseng, Y.H.: Generic Title Labeling for Clustered Documents. Expert Systems with Applications: An International Journal 37(3), 2247–2254 (Mar 2010)