Automatic Decomposition of Multi-Author Documents Using Grammar Analysis

Michael Tschuggnall and Günther Specht
Databases and Information Systems
Institute of Computer Science, University of Innsbruck, Austria
{michael.tschuggnall, guenther.specht}@uibk.ac.at

ABSTRACT

The task of text segmentation is to automatically split a text document into individual subparts, which differ according to specific measures. In this paper, an approach is presented that attempts to separate text sections of a collaboratively written document based on the grammar syntax of authors. The main idea is thereby to quantify differences of the grammatical writing style of authors and to use this information to build paragraph clusters, whereby each cluster is assigned to a different author. In order to analyze the style of a writer, text is split into single sentences, and for each sentence a full parse tree is calculated. Using the latter, a profile is computed subsequently that represents the main characteristics for each paragraph. Finally, the profiles serve as input for common clustering algorithms. An extensive evaluation using different English data sets reveals promising results, whereby a supplementary analysis indicates that in general common classification algorithms perform better than clustering approaches.

Keywords

Text Segmentation, Multi-Author Decomposition, Parse Trees, pq-grams, Clustering

1. INTRODUCTION

The growing amount of currently available data is hardly manageable without the use of specific tools and algorithms that provide relevant portions of that data to the user. While this problem is generally addressed with information retrieval approaches, another possibility to significantly reduce the amount of data is to build clusters. Within each cluster, the data is similar according to some predefined features. Thereby many approaches exist that propose algorithms to cluster plain text documents (e.g. [16], [22]) or specific web documents (e.g. [33]) by utilizing various features.

Approaches which attempt to divide a single text document into distinguishable units like different topics, for example, are usually referred to as *text segmentation* approaches. Here, also many features including statistical models, similarities between words or other semantic analyses are used. Moreover, text clusters are also used in recent plagiarism detection algorithms (e.g. [34]) which try to build a cluster for the main author and one or more clusters for intrusive paragraphs. Another scenario where the clustering of text is applicable is the analysis of multi-author academic papers: especially the verification of collaborated student works such as bachelor or master theses can be useful in order to determine the amount of work done by each student.

Using results of previous work in the field of intrinsic plagiarism detection [31] and authorship attribution [32], the assumption that individual authors have significantly different writing styles in terms of the syntax that is used to construct sentences has been reused. For example, the following sentence (extracted from a web blog): *"My chair started squeaking a few days ago and it's driving me nuts."* (S1) could also be formulated as *"Since a few days my chair is squeaking - it's simply annoying."* (S2) which is semantically equivalent but differs significantly according to the syntax as can be seen in Figure 1. The main idea of this work is to quantify those differences by calculating grammar profiles and to use this information to decompose a collaboratively written document, i.e., to assign each paragraph of a document to an author.

The rest of this paper is organized as follows: Section 2 at first recapitulates the principle of pq-grams, which represent a core concept of the approach. Subsequently the algorithm is presented in detail, which is then evaluated in Section 3 by using different clustering algorithms and data sets. A comparison of clustering and classification approaches is discussed in Section 4, while Section 5 depicts related work. Finally, a conclusion and future work directions are given in Section 6.

2. ALGORITHM

In the following the concept of pq-grams is explained, which serves as the basic stylistic measure in this approach to distinguish between authors. Subsequently, the concrete steps performed by the algorithm are discussed in detail.

2.1 Preliminaries: pq-grams

Similar to n-grams that represent subparts of given length n of a string, pq-grams extract substructures of an ordered, labeled tree [4]. The size of a pq-gram is determined by a stem (p) and a base (q) like it is shown in Figure 2. Thereby p defines how much nodes are included vertically, and q defines the number of nodes to be considered horizontally. For example, a valid pq-gram with $p = 2$ and $q = 3$ starting from `PP` at the left side of tree (S2) shown in Figure 1 would be `[PP-NP-DT-JJ-NNS]` (the concrete words are omitted).

The pq-gram index then consists of all possible pq-grams of a tree. In order to obtain all pq-grams, the base is shifted left and right additionally: If then less than p nodes exist horizontally, the corresponding place in the pq-gram is filled with $\star$, in-

In: G. Specht, H. Gamper, F. Klan (eds.): Proceedings of the 26^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 21.10.2014 - 24.10.2014, Bozen, Italy, published at http://ceur-ws.org.

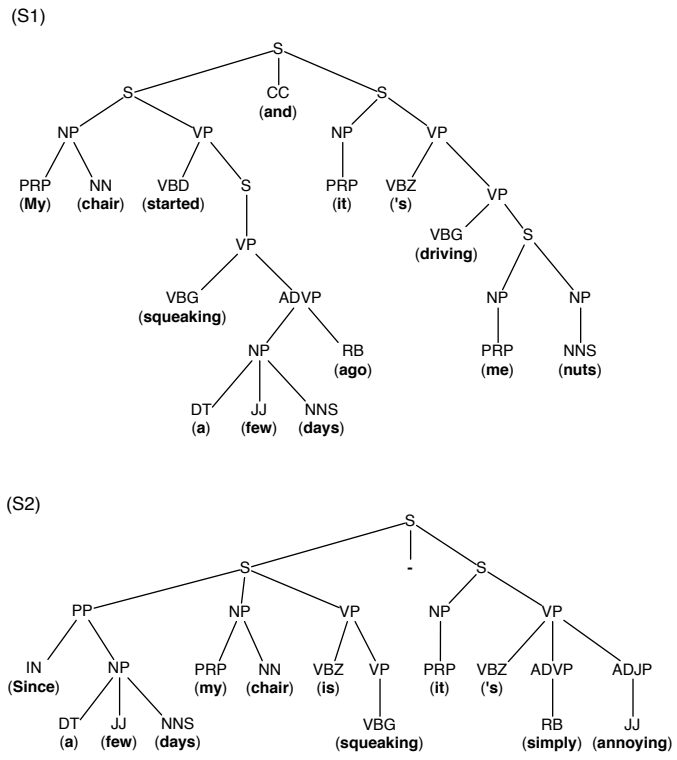

Figure 1: Grammar Trees of the Semantically Equivalent Sentences (S1) and (S2).

dicating a missing node. Applying this idea to the previous example, also the pq-gram `[PP-IN-*-*-*]` (no nodes in the base) is valid, as well as `[PP-NP-*-*-DT]` (base shifted left by two), `[PP-NP-*-DT-JJ]` (base shifted left by one), `[PP-NP-JJ-NNS-*]` (base shifted right by one) and `[PP-NP-NNS-*-*]` (base shifted right by two) have to be considered. As a last example, all leaves have the pq-gram pattern `[`*leaf_label*`-*-*-*-*]`.

Finally, the pq-gram index is the set of all valid pq-grams of a tree, whereby multiple occurrences of the same pq-grams are also present multiple times in the index.

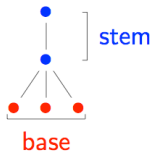

Figure 2: Structure of a pq-gram Consisting of Stem $p = 2$ and Base $q = 3$.

2.2 Clustering by Authors

The number of choices an author has to formulate a sentence in terms of grammar structure is rather high, and the assumption in this approach is that the concrete choice is made mostly intuitively and unconsciously. On that basis the grammar of authors is analyzed, which serves as input for common state-of-the-art clustering algorithms to build clusters of text documents or paragraphs. The decision of the clustering algorithms is thereby based on the frequencies of occurring pq-grams, i.e., on pq-gram profiles. In detail, given a text document the algorithm consists of the following steps:

1. At first the document is preprocessed by eliminating unnecessary whitespaces or non-parsable characters. For example, many data sets often are based on novels and articles of various authors, whereby frequently OCR text recognition is used due to the lack of digital data. Additionally, such documents contain problem sources like chapter numbers and titles or incorrectly parsed picture frames that result in non-alphanumeric characters.

2. Subsequently, the document is partitioned into single paragraphs. For simplification reasons this is currently done by only detecting multiple line breaks.

3. Each paragraph is then split into single sentences by utilizing a sentence boundary detection algorithm implemented within the *OpenNLP* framework[1]. Then for each sentence a full grammar tree is calculated using the *Stanford Parser* [19]. For example, Figure 1 depicts the grammar trees resulting from analyzing sentences (S1) and (S2), respectively. The labels of each tree correspond to a part-of-speech (POS) tag of the Penn Treebank set [23], where e.g *NP* corresponds to a noun phrase, *DT* to a determiner or *JJS* to a superlative adjective. In order to examine the building structure of sentences only like it is intended by this work, the concrete words, i.e., the leafs of the tree, are omitted.

4. Using the grammar trees of all sentences of the document, the pq-gram index is calculated. As shown in Section 2.1 all valid pq-grams of a sentence are extracted and stored into a pq-gram index. By combining all pq-gram indices of all sentences, a pq-gram profile is computed which contains a list of all pq-grams and their corresponding frequency of appearance in the text. Thereby the frequency is normalized by the total number of all appearing pq-grams. As an example, the five mostly used pq-grams using $p = 2$ and $q = 3$ of a sample document are shown in Table 1. The profile is sorted descending by the normalized occurrence, and an additional rank value is introduced that simply defines a natural order which is used in the evaluation (see Section 3).

Table 1: Example of the Five Mostly Used pq-grams of a Sample Document.

pq-gram	Occurrence [%]	Rank
`NP-NN-*-*-*`	2.68	1
`PP-IN-*-*-*`	2.25	2
`NP-DT-*-*-*`	1.99	3
`NP-NNP-*-*-*`	1.44	4
`S-VP-*-*-VBD`	1.08	5

5. Finally, each paragraph-profile is provided as input for clustering algorithms, which are asked to build clusters based on the pq-grams contained. Concretely, three different feature sets have been evaluated: (1.) the frequencies of occurences of each pq-gram, (2.) the rank of each pq-gram and (3.) a union of the latter sets.

[1] Apache OpenNLP, http://incubator.apache.org/opennlp, visited July 2014

2.3 Utilized Algorithms

Using the WEKA framework [15], the following clustering algorithms have been evaluated: K-Means [3], Cascaded K-Means (the number of clusters is cascaded and automatically chosen) [5], X-Means [26], Agglomerative Hierarchical Clustering [25], and Farthest First [9].

For the clustering algorithms *K-Means*, *Hierarchical Clustering* and *Farthest First* the number of clusters has been predefined according to the respective test data. This means if the test document has been collaborated by three authors, the number of clusters has also been set to three. On the other hand, the algorithms *Cascaded K-Means* and *X-Means* implicitly decide which amount of clusters is optimal. Therefore these algorithms have been limited only in ranges, i.e., the minimum and maximum number of clusters has been set to two and six, respectively.

3. EVALUATION

The utilization of pq-gram profiles as input features for modern clustering algorithms has been extensively evaluated using different documents and data sets. As clustering and classification problems are closely related, the global aim was to experiment on the accuracy of automatic text clustering using solely the proposed grammar feature, and furthermore to compare it to those of current classification techniques.

3.1 Test Data and Experimental Setup

In order to evaluate the idea, different documents and test data sets have been used, which are explained in more detail in the following. Thereby single documents have been created which contain paragraphs written by different authors, as well as multiple documents, whereby each document is written by one author. In the latter case, every document is treated as one (large) paragraph for simplification reasons.

For the experiment, different parameter settings have been evaluated, i.e., the pq-gram values p and q have been varied from 2 to 4, in combination with the three different feature sets. Concretely, the following data sets have been used:

- **Twain-Wells (T-W)**: This document has been specifically created for the evaluation of in-document clustering. It contains 50 paragraphs of the book *"The Adventures of Huckleberry Finn"* by Mark Twain, and 50 paragraphs of *"The Time Machine"* by H. G. Wells[2]. All paragraphs have been randomly shuffled, whereby the size of each paragraph varies from approximately 25 words up to 280 words.

- **Twain-Wells-Shelley (T-W-S)**: In a similar fashion a three-author document has been created. It again uses (different) paragraphs of the same books by Twain and Wells, and appends it by paragraphs of the book *"Frankenstein; Or, The Modern Prometheus"* by Mary Wollstonecraft Shelley. Summarizing, the document contains 50 paragraphs by Mark Twain, 50 paragraphs by H. G. Wells and another 50 paragraphs by Mary Shelley, whereby the paragraph sizes are similar to the Twain-Wells document.

- **The Federalist Papers (FED)**: Probably the mostly referred text corpus in the field of authorship attribution is a series of 85 political essays called "The Federalist Papers" written by John Jay, Alexander Hamilton and James Madison in the 18th century. While most of the authorships are undoubted, many works have studied and questioned the correct authorship of 12 disputed essays [24], which have been excluded in the experiment.

- **The PAN'12 competition corpus (PAN12)**: As a well-known, state-of-the-art corpus originally created for the use in authorship identification, parts[3] of the PAN2012 corpus [18] have been integrated. The corpus is composed of several fiction texts and split into several subtasks that cover small- and common-length documents (1800-6060 words) as well as larger documents (up to 13000 words) and novel-length documents (up to 170,000 words). Finally, the test setused in this evaluation contains 14 documents (paragraphs) written by three authors that are distributed equally.

3.2 Results

The best results of the evaluation are presented in Table 2, where the best performance for each clusterer over all data sets is shown in subtable (a), and the best configuration for each data set is shown in subtable (b), respectively. With an accuracy of 63.7% the K-Means algorithm worked best by using $p = 2, q = 3$ and by utilizing all available features. Interestingly, the X-Means algorithm also achieved good results considering the fact that in this case the number of clusters has been assigned automatically by the algorithm. Finally, the hierarchical cluster performed worst gaining an accuracy of nearly 10% less than K-Means.

Regarding the best performances for each test data set, the results for the manually created data sets from novel literature are generally poor. For example, the best result for the two-author document Twain-Wells is only 59.6%, i.e., the accuracy is only slightly better than the baseline percentage of 50%, which can be achieved by randomly assigning paragraphs into two clusters.[4] On the other hand, the data sets reused from authorship attribution, namely the FED and the PAN12 data set, achieved very good results with an accuracy of about 89% and 83%, respectively. Nevertheless, as the other data sets have been specifically created for the clustering evaluation, these results may be more expressive. Therefore a comparison between clustering and classification approaches is discussed in the following, showing that the latter achieve significantly better results on those data sets when using the same features.

Method	p	q	Feature Set	Accuracy
K-Means	3	2	All	**63.7**
X-Means	2	4	Rank	**61.7**
Farthest First	4	2	Occurrence-Rate	**58.7**
Cascaded K-Means	2	2	Rank	**55.3**
Hierarchical Clust.	4	3	Occurrence-Rate	**54.7**

(a) Clustering Algorithms

Data Set	Method	p	q	Feat. Set	Accuracy
T-W	X-Means	3	2	All	**59.6**
T-W-S	X-Means	3	4	All	**49.0**
FED	Farth. First	4	3	Rank	**89.4**
PAN12-A/B	K-Means	3	3	All	**83.3**

(b) Test Data Sets

Table 2: Best Evaluation Results for Each Clustering Algorithm and Test Data Set in Percent.

[2]The books have been obtained from the Project Gutenberg library, `http://www.gutenberg.org`, visited July 2014

[3]the subtasks A and B, respectively

[4]In this case X-Means dynamically created two clusters, but the result is still better than that of other algorithms using a fixed number of clusters.

4. COMPARISON OF CLUSTERING AND CLASSIFICATION APPROACHES

For the given data sets, any clustering problem can be rewritten as classification problem with the exception that the latter need training data. Although a direct comparison should be treated with caution, it still gives an insight of how the two different approaches perform using the same data sets. Therefore an additional evaluation is shown in the following, which compares the performance of the clustering algorithms to the performance of the the following classification algorithms: Naive Bayes classifier [17], Bayes Network using the K2 classifier [8], Large Linear Classification using LibLinear [12], Support vector machine using LIBSVM with nu-SVC classification [6], k-nearest-neighbors classifier (kNN) using $k = 1$ [1], and a pruned C4.5 decision tree (J48) [28]. To compensate the missing training data, a 10-fold cross-validation has been used for each classifier.

Table 3 shows the performance of each classifier compared to the best clustering result using the same data and pq-setting. It can be seen that the classifiers significantly outperform the clustering results for the Twain-Wells and Twain-Wells-Shelley documents. The support vector machine framework (LibSVM) and the linear classifier (LibLinear) performed best, reaching a maximum accuracy of nearly 87% for the Twain-Wells document. Moreover, the average improvement is given in the bottom line, showing that most of the classifiers outperform the best clustering result by over 20% in average. Solely the kNN algorithm achieves minor improvements as it attributed the two-author document with a poor accuracy of about 60% only.

A similar general improvement could be achieved on the three-author document Twain-Wells-Shelley as can be seen in subtable (b). Again, LibSVM could achieve an accuracy of about 75%, whereas the best clustering configuration could only reach 49%. Except for the kNN algorithm, all classifiers significantly outperform the best clustering results for every configuration.

Quite different comparison results have been obtained for the Federalist Papers and PAN12 data sets, respectively. Here, the improvements gained from the classifiers are only minor, and in some cases are even negative, i.e., the classification algorithms perform worse than the clustering algorithms. A general explanation is the good performance of the clustering algorithms on these data sets, especially by utilizing the Farthest First and K-Means algorithms.

In case of the Federalist Papers data set shown in subtable (c), all algorithms except kNN could achieve at least some improvement. Although the LibLinear classifier could reach an outstanding accuracy of 97%, the global improvement is below 10% for all classifiers. Finally, subtable (d) shows the results for PAN12, where the outcome is quite diverse as some classifiers could improve the clusterers significantly, whereas others worsen the accuracy even more drastically. A possible explanation might be the small data set (only the subproblems A and B have been used), which may not be suited very well for a reliable evaluation of the clustering approaches.

Summarizing, the comparison of the different algorithms reveal that in general classification algorithms perform better than clustering algorithms when provided with the same (pq-gram) feature set. Nevertheless, the results of the PAN12 experiment are very diverse and indicate that there might be a problem with the data set itself, and that this comparison should be treated carefully.

p	q	Algorithm	Max	N-Bay	Bay-Net	LibLin	LibSVM	kNN	J48
2	2	X-Means	57.6	77.8	82.3	85.2	86.9	62.6	85.5
2	3	X-Means	56.6	79.8	80.8	81.8	83.3	60.6	80.8
2	4	X-Means	57.6	76.8	79.8	82.2	83.8	58.6	81.0
3	2	X-Means	59.6	78.8	80.8	81.8	83.6	59.6	80.8
3	3	X-Means	53.5	76.8	77.8	80.5	82.3	61.6	79.8
3	4	X-Means	52.5	81.8	79.8	81.8	83.8	63.6	82.0
4	2	K-Means	52.5	86.9	83.3	83.5	84.3	62.6	81.8
4	3	X-Means	52.5	79.8	79.8	80.1	80.3	59.6	77.4
4	4	Farth. First	51.5	72.7	74.7	75.8	77.0	60.6	75.8
		average improvement		**24.1**	**25.0**	**26.5**	**27.9**	**6.2**	**25.7**

(a) Twain-Wells

p	q	Algorithm	Max	N-Bay	Bay-Net	LibLin	LibSVM	kNN	J48
2	2	K-Means	44.3	67.8	70.8	74.0	75.2	51.0	73.3
2	3	X-Means	38.3	65.1	67.1	70.7	72.3	48.3	70.2
2	4	X-Means	45.6	63.1	68.1	70.5	71.8	49.0	69.3
3	2	X-Means	45.0	51.7	64.1	67.3	68.8	45.6	65.4
3	3	X-Means	47.0	57.7	64.8	67.3	68.5	47.0	65.9
3	4	X-Means	49.0	67.8	67.8	70.5	72.5	46.3	68.3
4	2	X-Means	36.2	61.1	67.1	69.1	69.5	50.3	65.1
4	3	K-Means	35.6	53.0	63.8	67.6	70.0	47.0	66.6
4	4	X-Means	35.6	57.7	66.1	68.5	69.3	42.3	66.8
		average improvement		**18.7**	**24.8**	**27.7**	**29.0**	**5.6**	**26.0**

(b) Twain-Wells-Shelley

p	q	Algorithm	Max	N-Bay	Bay-Net	LibLin	LibSVM	kNN	J48
2	2	Farth. First	77.3	81.1	86.4	90.9	84.2	74.2	81.8
2	3	Farth. First	78.8	85.6	87.4	92.4	89.0	78.8	82.8
2	4	X-Means	78.8	89.4	92.4	90.9	87.3	89.4	85.9
3	2	K-Means	81.8	82.6	87.9	92.4	85.5	80.3	83.8
3	3	K-Means	78.8	92.4	92.4	92.4	86.4	81.8	83.8
3	4	Farth. First	86.4	84.8	90.9	97.0	85.8	81.8	85.6
4	2	Farth. First	86.6	81.8	89.4	87.9	83.3	77.3	84.1
4	3	Farth. First	89.4	85.6	92.4	89.4	85.8	80.3	83.3
4	4	Farth. First	84.8	86.4	90.9	89.4	85.8	84.8	83.6
		average improvement		**3.0**	**7.5**	**8.9**	**3.4**	**-1.6**	**1.3**

(c) Federalist Papers

p	q	Algorithm	Max	N-Bay	Bay-Net	LibLin	LibSVM	kNN	J48
2	2	K-Means	83.3	83.3	33.3	100.0	100.0	100.0	33.3
2	3	K-Means	83.3	83.3	33.3	100.0	100.0	100.0	33.3
2	4	K-Means	83.3	83.4	33.3	100.0	100.0	100.0	33.3
3	2	K-Means	83.3	75.0	33.3	91.7	91.7	100.0	33.3
3	3	K-Means	83.3	100.0	33.3	100.0	91.7	100.0	33.3
3	4	Farth. First	75.0	66.7	33.3	100.0	100.0	91.7	33.3
4	2	K-Means	83.3	91.7	33.3	91.7	75.0	91.7	33.3
4	3	K-Means	83.3	75.0	33.3	100.0	75.0	91.7	33.3
4	4	K-Means	83.3	75.0	33.3	100.0	83.4	83.4	33.3
		average improvement		**-0.9**	**-49.1**	**15.8**	**8.4**	**13.0**	**-49.1**

(d) PAN12-A/B

Table 3: Best Evaluation Results for each Clustering Algorithm and Test Data Set in Percent.

5. RELATED WORK

Most of the traditional document clustering approaches are based on occurrences of words, i.e., inverted indices are built and used to group documents. Thereby a unit to be clustered conforms exactly to one document. The main idea is often to compute topically related document clusters and to assist web search engines to be able to provide better results to the user, whereby the algorithms proposed frequently are also patented (e.g. [2]). Regularly applied concepts in the feature extraction phase are the term frequency tf, which measures how often a word in a document occurs, and the term frequency-inverse document frequency $tf - idf$, which measures the significance of a word compared to the whole document collection. An example of a classical approach using these techniques is published in [21].

The literature on cluster analysis within a single document to discriminate the authorships in a multi-author document like it is done in this paper is surprisingly sparse. On the other hand, many approaches exist to separate a document into paragraphs of different topics, which are generally called *text segmentation* problems. In this domain, the algorithms often perform vocabulary analysis in various forms like word stem repetitions [27] or word frequency models [29], whereby *"methods for finding the topic boundaries include sliding window, lexical chains, dynamic programming, agglomerative clustering and divisive clustering"* [7]. Despite the given possibility to modify these techniques to also cluster by authors instead of topics, this is rarely done. In the following some of the existing methods are shortly summarized.

Probably one of the first approaches that uses stylometry to automatically detect boundaries of authors of collaboratively written

text is proposed in [13]. Thereby the main intention was not to expose authors or to gain insight into the work distribution, but to provide a methodology for collaborative authors to equalize their style in order to achieve better readability. To extract the style of separated paragraphs, common stylometric features such as word/sentence lengths, POS tag distributions or frequencies of POS classes at sentence-initial and sentence-final positions are considered. An extensive experiment revealed that styolmetric features can be used to find authorship boundaries, but that there has to be done additional research in order to increase the accuracy and informativeness.

In [14] the authors also tried to divide a collaborative text into different single-author paragraphs. In contrast to the previously described handmade corpus, a large data set has been computationally created by using (well-written) articles of an internet forum. At first, different neural networks have been utilized using several stylometric features. By using 90% of the data for training, the best network could achieve an F-score of 53% for multi-author documents on the remaining 10% of test data. In a second experiment, only letter-bigram frequencies are used as distinguishing features. Thereby an authorship boundary between paragraphs was marked if the cosine distance exceeded a certain threshold. This method reached an F-score of only 42%, and it is suspected that letter-bigrams are not suitable for the (short) paragraphs used in the evaluation.

A two-stage process to cluster Hebrew Bible texts by authorship is proposed in [20]. Because a first attempt to represent chapters only by bag-of-words led to negative results, the authors additionally incorporated sets of synonyms (which could be generated by comparing the original Hebrew texts with an English translation). With a modified cosine-measure comparing these sets for given chapters, two core clusters are compiled by using the *ncut* algorithm [10]. In the second step, the resulting clusters are used as training data for a support vector machine, which finally assigns every chapter to one of the two core clusters by using the simple bag-of-words features tested earlier. Thereby it can be the case, that units originally assigned to one cluster are moved to the other one, depending on the prediction of the support vector machine. With this two-stage approach the authors report a good accuracy of about 80%, whereby it should be considered that the size of potential authors has been fixed to two in the experiment. Nevertheless, the authors state that their approach could be extended for more authors with less effort.

6. CONCLUSION AND FUTURE WORK

In this paper, the automatic creation of paragraph clusters based on the grammar of authors has been evaluated. Different state-of-the-art clustering algorithms have been utilized with different input features and tested on different data sets. The best working algorithm K-Means could achieve an accuracy of about 63% over all test sets, whereby good individual results of up to 89% could be reached for some configurations. On the contrary, the specifically created documents incorporating two and three authors could only be clustered with a maximum accuracy of 59%.

A comparison between clustering and classification algorithms using the same input features has been implemented. Disregarding the missing training data, it could be observed that classifiers generally produce higher accuracies with improvements of up to 29%. On the other hand, some classifiers perform worse on average than clustering algorithms over individual data sets when using some pq-gram configurations. Nevertheless, if the maximum accuracy for each algorithm is considered, all classifiers perform significantly better as can be seen in Figure 3. Here the best performances of all utilized classification and clustering algorithms are illustrated. The

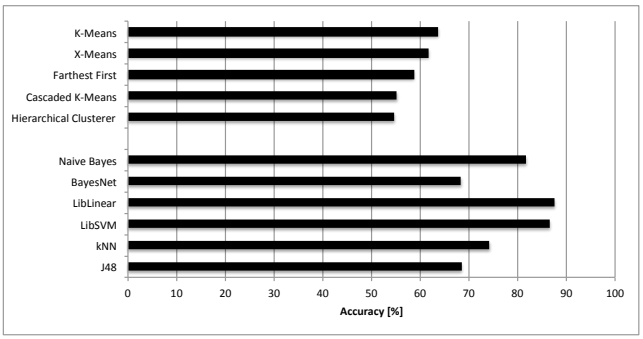

Figure 3: Best Evaluation Results Over All Data Sets For All Utilized Clustering and Classification Algorithms.

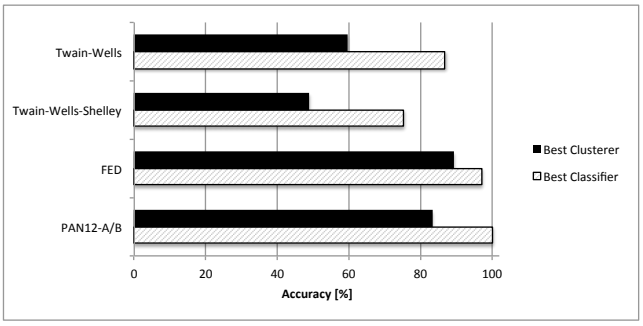

Figure 4: Best Clustering and Classification Results For Each Data Set.

linear classification algorithm LibLinear could reach nearly 88%, outperforming K-Means by 25% over all data sets.

Finally, the best classification and clustering results for each data set are shown in Figure 4. Consequently the classifiers achieve higher accuracies, whereby the PAN12 subsets could be classified 100% correctly. As can be seen, a major improvement can be gained for the novel literature documents. For example, the best classifier reached 87% on the Twain-Wells document, whereas the best clustering approach achieved only 59%.

As shown in this paper, paragraphs of documents can be split and clustered based on grammar features, but the accuracy is below that of classification algorithms. Although the two algorithm types should not be compared directly as they are designed to manage different problems, the significant differences in accuracies indicate that classifiers can handle the grammar features better. Nevertheless future work should focus on evaluating the same features on larger data sets, as clustering algorithms may produce better results with increasing amount of sample data.

Another possible application could be the creation of whole document clusters, where documents with similar grammar are grouped together. Despite the fact that such huge clusters are very difficult to evaluate - due to the lack of ground truth data - a navigation through thousands of documents based on grammar may be interesting like it has been done for music genres (e.g. [30]) or images (e.g. [11]). Moreover, grammar clusters may also be utilized for modern recommendation algorithms once they have been calculated for large data sets. For example, by analyzing all freely available books from libraries like Project Gutenberg, a system could recommend other books with a similar style based on the users reading history. Also, an enhancement of current commercial recommender systems that

are used in large online stores like Amazon is conceivable.

7. REFERENCES

[1] D. Aha and D. Kibler. Instance-Based Learning Algorithms. *Machine Learning*, 6:37–66, 1991.

[2] C. Apte, S. M. Weiss, and B. F. White. Lightweight Document Clustering, Nov. 25 2003. US Patent 6,654,739.

[3] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[4] N. Augsten, M. Böhlen, and J. Gamper. The pq-Gram Distance between Ordered Labeled Trees. *ACM Transactions on Database Systems (TODS)*, 2010.

[5] T. Caliński and J. Harabasz. A Dendrite Method for Cluster Analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 1974.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[7] F. Y. Choi. Advances in Domain Independent Linear Text Segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33. Association for Computational Linguistics, 2000.

[8] G. F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks From Data. *Machine learning*, 9(4):309–347, 1992.

[9] S. Dasgupta. Performance Guarantees for Hierarchical Clustering. In *Computational Learning Theory*, pages 351–363. Springer, 2002.

[10] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: Spectral Clustering and Normalized Cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.

[11] A. Faktor and M. Irani. "Clustering by Composition" - Unsupervised Discovery of Image Categories. In *Computer Vision–ECCV 2012*, pages 474–487. Springer, 2012.

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[13] A. Glover and G. Hirst. Detecting Stylistic Inconsistencies in Collaborative Writing. In *The New Writing Environment*, pages 147–168. Springer, 1996.

[14] N. Graham, G. Hirst, and B. Marthi. Segmenting Documents by Stylistic Character. *Natural Language Engineering*, 11(04):397–415, 2005.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: an Update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[16] A. Hotho, S. Staab, and G. Stumme. Ontologies Improve Text Document Clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 541–544. IEEE, 2003.

[17] G. H. John and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[18] P. Juola. An Overview of the Traditional Authorship Attribution Subtask. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[19] D. Klein and C. D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003.

[20] M. Koppel, N. Akiva, I. Dershowitz, and N. Dershowitz. Unsupervised Decomposition of a Document into Authorial Components. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1356–1364, Stroudsburg, PA, USA, 2011.

[21] B. Larsen and C. Aone. Fast and Effective Text Mining Using Linear-Time Document Clustering. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22. ACM, 1999.

[22] Y. Li, S. M. Chung, and J. D. Holt. Text Document Clustering Based on Frequent Word Meaning Sequences. *Data & Knowledge Engineering*, 64(1):381–404, 2008.

[23] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330, June 1993.

[24] F. Mosteller and D. Wallace. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, 1964.

[25] F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms. *The Computer Journal*, 26(4):354–359, 1983.

[26] D. Pelleg, A. W. Moore, et al. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *ICML*, pages 727–734, 2000.

[27] J. M. Ponte and W. B. Croft. Text Segmentation by Topic. In *Research and Advanced Technology for Digital Libraries*, pages 113–125. Springer, 1997.

[28] J. R. Quinlan. *C4.5: Programs for Machine Learning*, volume 1. Morgan Kaufmann, 1993.

[29] J. C. Reynar. Statistical Models for Topic Segmentation. In *Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 357–364, 1999.

[30] N. Scaringella, G. Zoia, and D. Mlynek. Automatic Genre Classification of Music Content: a Survey. *Signal Processing Magazine, IEEE*, 23(2):133–141, 2006.

[31] M. Tschuggnall and G. Specht. Using Grammar-Profiles to Intrinsically Expose Plagiarism in Text Documents. In *Proc. of the 18th Conf. of Natural Language Processing and Information Systems (NLDB)*, pages 297–302, 2013.

[32] M. Tschuggnall and G. Specht. Enhancing Authorship Attribution By Utilizing Syntax Tree Profiles. In *Proc. of the 14th Conf. of the European Chapter of the Assoc. for Computational Ling. (EACL)*, pages 195–199, 2014.

[33] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proc. of the 21st annual international ACM conference on Research and development in information retrieval (SIGIR)*, pages 46–54. ACM, 1998.

[34] D. Zou, W.-J. Long, and Z. Ling. A Cluster-Based Plagiarism Detection Method. In *Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September*, 2010.