Hierarchy Identification for Automatically Generating Table-of-Contents

Nicolai Erbs[α] **Iryna Gurevych**[αβ] **Torsten Zesch**[γ]

[α]Ubiquitous Knowledge Processing Lab Department of Computer Science, Technische Universität Darmstadt

[β]Information Center for Education German Institute for Educational Research and Educational Information

[γ]Language Technology University of Duisburg-Essen

Abstract

A table-of-contents (TOC) provides a quick reference to a document's content and structure. We present the first study on identifying the hierarchical structure for automatically generating a TOC using only textual features instead of structural hints e.g. from HTML-tags. We create two new datasets to evaluate our approaches for hierarchy identification. We find that our algorithm performs on a level that is sufficient for a fully automated system. For documents without given segment titles, we extend our work by automatically generating segment titles.

We make the datasets and our experimental framework publicly available in order to foster future research in TOC generation.

1 Introduction

A table-of-contents (TOC) provides an easy way to gain an overview about a document as a TOC presents the document's content and structure. At the same time, a TOC captures the relative importance of document topics by arranging the topic titles in a hierarchical manner. Thus, TOCs might be used as a short document summary that provides more information about search results in a search engine. Figure 1 provides a sketch of such a search interface. Instead of a thumbnail of the document like most search engines, or a clustering of search results (Carpineto et al., 2009), we propose to use an automatically extracted TOC.

The task of automatically generating a table-of-contents can be tackled with the subtasks document segmentation, segment title generation, and hierarchy identification. The first step splits the document into topical parts, the second step generates an informative title for each segment, and

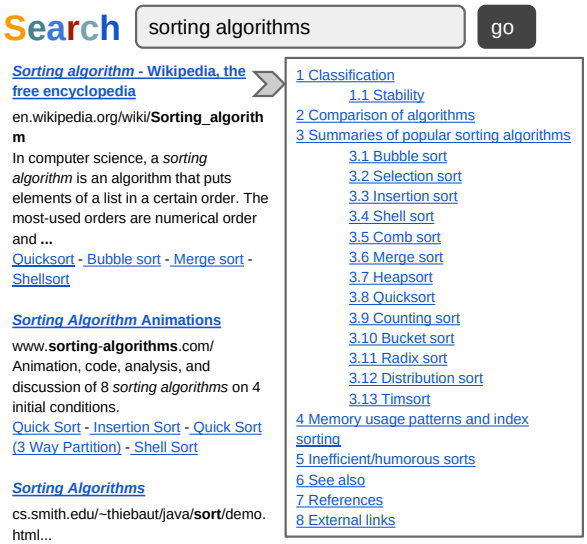

Figure 1: Search user interface showing a TOC along with the search results.

the third step decides whether a segment is on a higher, equal, or lower level than the previous segment. This paper presents novel approaches for the third subtask: hierarchy identification. Additionally, it presents a detailed analysis of results for segment title generation on the presented datasets.

Many documents are already segmented but only few documents already contain an explicit hierarchical TOC (e.g. Wikipedia articles), while for most documents it needs to be automatically identified. For some documents, identification is straight-forward, e.g. if an HTML document already contains hierarchically structured headlines (`<h1>`, `<h2>`, etc). We focus on the most challenging case in which only the textual content of the documents' segments are available and the hierarchy needs to be inferred using Natural Language Processing.

We present a framework for automatically identifying the hierarchy of two segments based on semantic and lexical features. We perform linguistic

Proceedings of Recent Advances in Natural Language Processing, pages 252–260,
Hissar, Bulgaria, 7-13 September 2013.

Contents

Figure 2: TOC of this paper

preprocessing including named entity recognition (Finkel et al., 2005), keyphrase extraction (Mihalcea and Tarau, 2004), and chunking (Schmid, 1994) which are then used as features for machine learning.

To foster future research, we present two new datasets and compare results on these datasets and the one presented by Branavan et al. (2007).

Our research contribution is to develop new algorithms for segment hierarchy identification, to present new evaluation datasets for all subtasks, and to compare our newly developed methods with the state of the art. We also provide a comprehensive analysis of the benefits and shortcomings of the applied methods. Figure 2 gives an overview of the paper's organization (and at the same time highlights the usefulness of a TOC for the reader). Thus, we may safely skip the enumeration of paper sections and their content that usually concludes the introduction.

2 Related Work

For some documents, the hierarchy of segments can be induced using HTML-based features. Pembe and Güngör (2010) focus on DOM tree and formatting features, but also use occurrences of manually crafted cue phrases such as *back to top*. However, most features are only applicable in very few cases where HTML markup directly provides a hierarchy. In order to provide a uniform user experience, a TOC also needs to be generated for documents where HTML-based methods fail or when only the textual content is available.

Feng et al. (2005) train a classifier to detect semantically coherent areas on a page. However, they make use of the existing HTML markup and return areas of the document instead of identifying hierarchical structures for segments. Besides markup and position features, they use features based on unigrams and bigrams for classifying a segment into one of 12 categories.

For segment title generation we divide related work into the following classes:

Text-based approaches make use of only the text in the corresponding segment. Therefore, titles are limited to words appearing in the text. They can be applied in all situations, but will often create trivial or even wrong titles.

Supervised approaches learn a model of which document segments usually have a certain title. They are highly precise, but require training data and are limited to an *a priori* determined set of titles for which the model is trained.

In the following, we organize the few available previous papers on this topic according to these two classes. The text-based approach by Lopez et al. (2011) uses a position heuristic. Each noun phrase in a segment is given a score depending on its position and its tf.idf value.

The supervised approach by Branavan et al. (2007) trains an incremental perceptron algorithm (Collins and Roark, 2004; Daumé and Marcu, 2005) to predict titles. It uses rules based on the hierarchical structure of the document[1] to rerank the candidates towards the best global solution. Nguyen and Shimazu (2009) expand the supervised approach by Branavan et al. (2007) using word clusters as additional features. Both approaches are trained and tested on the Cormen dataset. The book is split into a set of 39 independent documents at boundaries of segments of the second level. The newly created documents are randomly selected for training (80%) and testing (20%). Such an approach is not suited for our scenario of end-to-end TOC creation, as we want to generate a TOC for a whole document and cannot train on parts of it. Besides, this tunes the system towards special characteristics of the book instead of having a domain-independent system.

Keyphrase extraction methods (Frank et al., 1999; Turney, 2000) may also be used for segment title generation if a reader prefers even shorter headlines. These methods can be either text-based or supervised.

[1] E.g. neighboring segments must not have the same title.

3 Experimental Setup

Our system tackles the problem using a supervised classifier predicting the relation between the segments. Two segments can be on the *same*, *higher*, or *lower* level. Formally, the difference of a segment with level l_0 and a following segment with level l_1 is any integer n $\in [-\infty..\infty]$ for which n$= l_1 - l_0$. However, our analysis on the development data has shown that n typically is in the range of $\in [-2..2]$ which means that a following segment is at most 2 levels higher or lower than the previous segment.

We identified the following categories of features that solely make use of the text in each segment (we refer to these features as in-document features):

N-gram features We identify the top-500 n-grams in the collection and use them as Boolean features for each segment. The feature value is set to `true` if the n-gram appears, `false` otherwise. These features reflect reoccurring cue phrases and generic terms for fixed segments like the introduction.

Length-based We compute the number of characters (including whitespaces) for both segments and use their difference as feature value. We apply the same procedure for the number of tokens and sentences. A higher-level segment might be shorter because it provides a summary of the following more detailed segments.

Entity-based We identify all named entities in each segment and return a Boolean feature if they share at least one entity. This feature is based on the assumption that two segments having the same entities are related. Two related segments are more likely on the same level or the second segment is a lower-level segment.

Noun chunk features All noun chunks in both segments are identified using the TreeTagger (Schmid, 1994) and then the average number of tokens for each of the segments is computed. The feature value is the difference of the average phrase length. Phrases in lower-level segments are longer because they are more detailed. In the example from Figure 1, the term *bubble sort algorithm* is longer than the frequently occurring upper level phrase *sorting algorithm*.

Additionally, the number of chunks that appear in both segments is divided by the number of chunks that appear in the second segment. If a term like *sorting algorithm* is the only shared term in both segments and the second segment contains in total ten phrases, then the noun chunk overlap is 10%. This feature is based on the assumption that lower-level segments mostly mention noun chunks that have been already introduced earlier.

Keyphrase-based We apply the state-of-the-art keyphrase extraction approach TextRank (Mihalcea and Tarau, 2004) and identify a ranked list of keyphrases in each segment. We compare the top-k (k $\in [1, 2, 3, 4, 5, 10, 20]$) keyphrases of each segment pair and return `true` if at least one keyphrase appears in both segments. These features also reflect topically related segments.

Frequency We apply another feature set which uses a background corpus in addition to the text of the segments. We use the Google Web1T corpus (Brants and Franz, 2006) to retrieve the frequency of a term. The average frequency of the top-k (k $\in [5, 10]$) keyphrases in a segment is calculated and the difference between two segments is the feature value. We expect lower-level segments to contain keyphrases that are less frequently used.

We use WEKA (Hall et al., 2009) to train the classifier and report results obtained with SVM, which performed best on the development set.[2] We evaluate all approaches by computing the accuracy as the fraction of correctly identified hierarchy relations. As a baseline, we consider all segments to be on the same level.

3.1 Datasets

Branavan et al. (2007) extracted a single TOC from an algorithms textbook (Cormen et al., 2001) and split it into a training and a test set. We use the complete TOC as a test set and refer to it as *Cormen*. As a single TOC is a shallow basis for experimental results, we create two additional datasets

[2] We experimented with Naïve Bayes and J48 but results were significantly lower.

Name	doc	seg	$\varnothing \frac{tok}{seg}$
Cormen	1	607	733
Gutenberg	18	1,312	1927
Wikipedia	277	3,680	399

Table 1: Characteristics of evaluation datasets. Showing the total number of documents (doc), segments (seg) and average number of tokens in each segment ($\varnothing \frac{tok}{seg}$).

	Hierarchy level				
Name	1	2	3	4	5
Cormen	.00	.02	.08	.41	.48
Wikipedia	.07	.48	.41	.04	.00
Gutenberg	.01	.35	.49	.12	.03

Table 2: Distribution of segments over levels of the evaluation corpora.

containing real-world tables of contents, allowing us to evaluate on different domains and styles of hierarchies.

We create the first dataset from randomly selected featured articles in Wikipedia. They have been shown to be of high quality (Stein and Hess, 2007) and are complex enough to contain hierarchical TOCs. We create a second dataset using 55 books from the project Gutenberg.[3] We refer to these datasets as *Wikipedia* and *Gutenberg*. We annotated these datasets with the hierarchy level of each segment, ranging from 1 (top-level segment) to the lowest-level segment found in the datasets.

Table 1 gives an overview of the datasets regarding the segment structure. Although the Cormen dataset consists of one book only, it contains more segments than an average document in any other dataset and thus is a valuable evaluation resource. The Wikipedia dataset contains on average the fewest tokens in each segment, in other words – the most fine-grained TOC. The Wikipedia and Gutenberg dataset cover a broad spectrum of topics while the Cormen dataset is focused on computational algorithms.

Table 2 shows the distribution of levels in the datasets. The *Cormen* dataset has a much deeper structure compared to the other two datasets. The fraction of segments on the first level is below 1% because a single document may have only one top-level segment and this document contains far more than 100 segments. This is a special characteristic of this book: since it is often used to quickly look up specific topics, the authors provide a very fine-grained table-of-contents. In *Wikipedia*, most of the segments are on the second level. Articles in Wikipedia are rather short, because according to the Wikipedia author guidelines a segment of a Wikipedia article is moved into an independent article if it gets too long. The *Gutenberg* dataset is more balanced as it contains documents from different authors. Similar to the Wikipedia dataset, most segments are on the second and third level.

[3]The same collection of books was used by Csomai and Mihalcea (2006) for experiments on back-of-the-book indexing. They mostly cover the domains humanities, science, and technology.

	Pairwise hierarchy relation				
Name	n= 2	n= 1	n= 0	n= −1	n= −2
Cormen	.00	.20	.60	.16	.03
Wikipedia	.00	.15	.71	.13	.01
Gutenberg	.00	.10	.80	.09	.01

Table 3: Distribution of pairwise level difference of segments of the evaluation corpora.

We focus on the pairwise classification in this paper and investigate the pairwise relation of neighboring segments. Two segments on the same level have a hierarchy relation of n=0, a segment that is one level lower has a hierarchy relation of n=1. Table 3 shows that for all datasets most of the segment pairs (neighboring segments) are on the same level. Although there are segments which are two level higher or three levels higher than the previous segment, this is the case for no more than 1% of all segment pairs. The Cormen has the highest deviation of level relation. This is due to the fact that its segments have a broad distribution of levels (see Table 2). Segments in the Gutenberg dataset, on the other hand, are in 80% of all cases on the same level as the previous segment. The case that the next segment is two level lower, i.e. n=2, is very unlikely. This is in line with our expectations that a writer does not skip levels when starting a lower level segment.

4 Experiments and Results

We evaluate performance of our system using 10-fold cross-validation on previously unseen data using The Lab as experimental framework (Eckart de Castilho and Gurevych, 2011). Performance is measured in terms of accuracy and is defined as the ratio of correctly identified relations.

Table 4 shows our results on each dataset. Always predicting two segments to be on the same level is a strong baseline, as this is the case for

	Cormen	WP	Gutenb.
Baseline (*always equal)*	.60	.71	.80
(1) N-gram features	**.86**	.64	.86
(2) Length features	.62	.76	.80
(3) Entity features	.60	.71	.80
(4) Noun chunk features	.83	**.86**	**.91**
(5) Keyphrase features	.60	.71	.80
(6) Frequency features	.60	.71	.80
All features	.86	.77	.86
All features w/o (1)	.83	**.86**	**.91**
All features w/o (3) & (5)	**.87**	.77	.86

Table 4: Accuracy of approaches for hierarchy identification. Best results of feature groups and combinations are marked bold.

Actual \ Predicted	2	1	0	−1	−2
2	**-**	4	-	-	-
1	-	**567**	-	-	-
0	-	-	**2,585**	-	-
−1	-	-	478	**-**	-
−2	-	-	24	-	**-**

Table 5: Confusion matrix for best system (all features w/o n-gram features) on Wikipedia dataset. Correctly identified segments are marked bold.

60.2% of cases in the Cormen and 79.8% of cased in the Gutenberg dataset. The table shows results for each of the feature groups defined in Section 3 numbered from (1) to (6). N-gram features perform best on the Cormen dataset while they perform worse than the baseline on the Wikipedia (WP) dataset. This difference might be due to the topic diversity in the Wikipedia and Cormen datasets. Wikipedia covers many topics, while Cormen is focused on a single topic (algorithms) and thus containing reappearing n-grams.

Noun chunk features are the best-performing group of features on the Wikipedia and Gutenberg and second best on the Cormen dataset. Entity, keyphrase, and frequency features do not improve the baseline in any of the presented datasets. Apparently, they are no good indicator for the hierarchical structure of document segments.

Combining all features further improves results on the Cormen dataset. However, the best results are obtained by combining all besides entity and keyphrase features. On the other two datasets (Wikipedia and Gutenberg), a combination of all features decreases accuracy compared to a supervised system using only noun chunk features. The highest accuracy is obtained by using all features besides n-gram features.

Based on our observation that a combination

Actual \ Predicted	2	1	0	−1	−2
2	**-**	4	-	-	-
1	-	**539**	17	11	-
0	-	14	**2,115**	455	1
−1	-	1	323	**154**	-
−2	-	-	12	12	**-**

Table 6: Confusion matrix for a system using all features on Wikipedia dataset. Correctly identified segments are marked bold.

of all features performs worse than a selection of features, we analyzed the confusion matrix of the corresponding systems. Table 5 shows the confusion matrix for the best performing system from Table 4 on the Wikipedia dataset using selected features (all w/o n-gram features). The system is optimized towards accuracy and trained on unbalanced training data. This leads to a system returning either n$=1$ (next level is one level lower) or n$=0$ (same level). There are no cases where a lower-level segment is incorrectly classified as a higher-level segment but all cases with $|n| \geq 2$ are incorrectly classified as having a level difference of one.

Table 6 shows the confusion matrix for a system using all features on the same dataset as before (Wikipedia). The system also covers the case n$=-1$ (next level is one level higher), thus creating more realistic TOCs. In contrast to the previous system (see Table 5), some higher-level segment relations (n<0) are incorrectly classified as lower-level segment relations (n>0). Although the system using all features returns a lower precision than the one using selected features, it better captures the way writers construct documents (also having segments on a higher level than previous segments).

Overall, results show that automatic hierarchy identification provides a TOC with a sufficient quality. To support this observation, Figure 3 shows the correct and predicted TOCs for the article about *Apollo 8* from the Wikipedia dataset. The correct TOC is on the left and the predicted TOC is on the right.

Section 1.3 (*Mission control*) was erroneously identified as being on a higher level than the previous section. The system fails to identify that both segments are about the crew (backup and mission control crew). The section *Planning* is correctly identified as having a higher level than the previous segment but leading to a different numbering

Correct TOC	Predicted TOC
1 Crew	1 Crew
1.1 Backup crew	1.1 Backup crew
1.2 Mission control	2 Mission control
1.3 Mission insignia	2.1 Mission insignia
2 Planning	3 Planning
3 Saturn V	4 Saturn V
4 Mission	5 Mission
4.1 Parameter summary	5.1 Parameter summary
4.2 Launch and trans-lunar injection	5.2 Launch and trans-lunar injection
4.3 Lunar trajectory	5.3 Lunar trajectory
4.4 Lunar sphere of influence	5.4 Lunar sphere of influence
4.5 Lunar orbit	5.5 Lunar orbit
4.5.1 Earthrise	5.5.1 Earthrise
4.6 Unplanned manual re-alignment	5.5.2 Unplanned manual re-alignment
4.7 Cruise back to Earth and re-entry	5.6 Cruise back to Earth and re-entry
5 Historical importance	6 Historical importance
6 Spacecraft location	7 Spacecraft location
7 In film	8 In film

Figure 3: Correct and predicted TOCs of article about Apollo 8 from the Wikipedia dataset.

(5 instead of 4 due to earlier errors). Not all of the remaining segment relations are correctly identified but the overall TOC still provides a quick reference of the article's content. It allows a reader to quickly decide whether the article about *Apollo 8* fulfills his information need.

5 Segment Title Generation

So far, we have shown that our system is able to automatically predict a TOC for documents segment boundaries. In order to extend our system to documents that do not have titles for segments, we add a segment title generation step. News documents are very often segmented into smaller parts, but usually do not contain segment titles.[4]

We decided not to reuse existing datasets from summarization or keyphrase extraction tasks, as they are only focused on one possible style of titles (i.e. summaries or keyphrases). Instead, we apply our algorithms to the previously presented datasets for hierarchy identification (see Section 3.1) and analyze their characteristics with respect to their segment titles. The percentage of titles that actually appear in the corresponding segments is lowest for the Wikipedia dataset (18%) while it is highest on the Cormen dataset (27%). In the Gutenberg dataset 23% of all titles appear in the text. The high value for the Cormen dataset is due to the specific characteristic that segment titles are repeated very often at the beginning of a segment.[5]

[4]For example, `cnn.com` uses *story paragraphs*.

[5]For example, the segment *Quicksort* begins with: *Quicksort is a sorting algorithm ...*

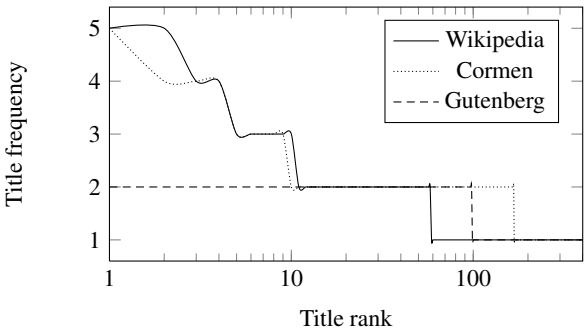

Figure 4: Frequency distribution of a random sample of 607 titles on log-log-scale: it follows a power-law distribution.

Frequency Distribution of Titles We further analyze the datasets in terms of segment counts for each title. Figure 4 shows the frequency of titles in the evaluation set on a logarithmic scale. We choose a random sample of 607 titles, which is the lowest number of titles in all three corpora, to allow a fair comparison across corpora. For all three datasets, most titles are used for few segments. For the datasets Wikipedia and Cormen some titles are used more frequently. In comparison to that, the most-frequent title of the Gutenberg dataset appears twice, only. Thus, we expect the supervised approaches to be most beneficial on the Wikipedia dataset. On the Cormen dataset we cannot apply any supervised approaches due to the lack of training data.

5.1 Experimental Setup

Text-based approaches As simple baselines, we use the first token and the first noun phrase occurring in each segment. As a more sophisticated baseline, we rank tokens according to their tf–idf scores. Additionally, we use TextRank (Mihalcea and Tarau, 2004) to rank noun phrases according to their co-occurrence frequencies.

As named entities from a segment are often used as titles, we extract them using the Stanford Named Entity Tagger (Finkel et al., 2005) and take the first one as the segment title.[6]

Supervised approaches We train a text classification model based on character 6-grams.[7] for

[6]We also experimented using the most frequent entity but achieved lower results.

[7]A previous evaluation has shown that 6-grams yield the best results for this task on all development sets. We used LingPipe: `http://alias-i.com/lingpipe` for classification.

each of the most frequent titles in each dataset. In Wikipedia, most articles have sections like *See also*, *References*, or *External links*, while books usually start with a chapter *Preface*. We restrict the list of title candidates to those appearing at least twice in the training data. We use a statistical model for predicting the title of a segment

In contrast to previous approaches (Branavan et al., 2007; Nguyen and Shimazu, 2009; Jin and Hauptmann, 2001), we do not train on parts of the same document for which we want to predict titles, but rather on full documents of the same type (Wikipedia articles and books). This is an important difference, as in our usage scenario we need to generate full TOCs for previously unseen documents. On the Cormen dataset we cannot perform a trainings phase as it consists of one book.

Evaluation Metrics We evaluated all approaches using two evaluation metrics. We propose **accuracy** as evaluation metric. A generated title is counted as correct only if it exactly matches the correct title. Hence, methods that generate long titles by adding many important phrases are penalized.

The **Rouge** evaluation metric is commonly used for evaluating summarization systems. It is based on n-gram overlap, where —in our case— the generated title is compared to the gold title. We use Rouge-L which is based on the longest common subsequence. This metric is frequently used in previous work for evaluating supervised approaches to generating TOCs because it considers near misses. We believe that it is not well suited for evaluating title generation, however, we use it for the sake of comparison with related work.

5.2 Experiments and Results

Table 7 shows the results of title generation approaches on the three datasets. On the Cormen dataset, we compare our approaches with two state-of-the-art methods. For the newly created datasets no previous results are available.

Using the first noun phrase returns the best titles on the Cormen dataset, which is in agreement with our observation from Section 5.1 that many segments repeat their title in the beginning. This also explains the high performance of the state-of-the-art approaches which are also taking the position and part of speech of candidates into account. Branavan et al. (2007) report about a feature for the supervised systems eliminating generic phrases without giving example of these phrases.

Supervised text classification approach works quite well in case of the Wikipedia dataset with its frequently appearing titles. The approach does not work well on the Gutenberg dataset, as segments such as *Preface* treat different topics in most Gutenberg books. Consequently, the text classifier is not able to learn the specific properties of that segment. In future work, it will be necessary to adapt the classifier in order to focus on non-standard features that better grasp the function of a segment inside a document. For example, the introduction of a scientific paper always reads "introduction-like" while the covered topic changes from paper to paper. This is in line with research concerning topic bias (Mikros and Argiri, 2007; Brooke and Hirst, 2011) in which topic-independent features are applied.

The overall level of performance in terms of accuracy and Rouge seems rather low. However, accuracy is only a rough estimate of the real performance, as many good titles might not be represented in the gold standard and Rouge is higher when comparing longer texts. Besides, a user might be interested in a specialized table-of-contents, such as one consisting only of named entities. For example, in a document about US presidential elections, a TOC consisting only of the names of presidents might be more informative than one consisting of the dates of the four-year periods. A flexible system for generating segment titles enables the user to decide on which titles are more interesting and thus increasing the user's benefit.

Combination of approaches As we have discussed, the usage of titles highly depends on the domain of the document and the expectations of the reader. We aim to overcome the limitations of single approaches by combining multiple approaches and integrating the reader's choice to improve the overall acceptance of a title generation system. It is essential that a combination reflects different styles of titles to cover most of the reader's preferences.

We combine complementary approaches based on three baseline systems (first NP, tf–idf, and named entities) and additionally the supervised approach (text classification). We expect the three text-based features to provide a stable performance, while the supervised approach may boost

Approach	Type	Wikipedia Acc.	Wikipedia Rouge-L	Gutenberg Acc.	Gutenberg Rouge-L	Cormen Acc.	Cormen Rouge-L
(Branavan et al., 2007)	*Supervised*	-	-	-	-	-	*.249*
(Nguyen and Shimazu, 2009)		-	-	-	-	-	*.281*
First token	Baselines	.007	.034	.004	.078	.010	.137
First NP		.012	.112	.037	**.180**	**.061**	**.364**
tf–idf		.017	.057	**.042**	.094	.020	.206
TextRank	Text	.014	.058	.011	.060	.012	.195
Named entity		.006	.046	.011	.065	.000	.037
Text classification	Supervised	**.133**	**.169**	.004	.008	*	*
First NP, tf–idf, named entity	Combination	.034	n/a	.069	n/a	.076	n/a
+ Text classification		.168	n/a	.072	n/a	.077	n/a

Table 7: Title generation results. No results for supervised text classification on the Cormen dataset are shown since no training data is available.

the performance on some datasets. As these approaches typically use an independent set of title candidates, they can potentially achieve a higher performance. Commonly used combination strategies like *voting* or complex strategies (Chen, 2011) can only be applied within approaches from the same class, as different classes will output different titles. Besides, it is desirable to create a diversity of candidates without ignoring titles generated by only one approach.

Results in Table 7 reveals that a combination of approaches provides the highest accuracy of all approaches. We cannot compare a list of generated titles to a gold title with Rouge, thus not presenting any numbers (n/a). We utilize the benefit of accuracy allowing to compare a set of generated titles to a gold title. In a real-world setting, a user selects the best title from the list which means that only one suggestion has to match the gold standard. Although providing a larger result set increases accuracy, results are stable for all datasets.

6 Conclusions and Future Work

We presented the first study on automatically identifying the hierarchical structure of a table-of-contents for different kinds of text (articles and books from different domains). The task of *segment hierarchy identification* is a new task which has not been investigated for non-HTML text. We created two new evaluation datasets for this task, and used a supervised approach based on textual features and a background corpus and significantly improved results over a strong baseline. For documents with missing segment titles, *generating segment titles* is an interesting use case for keyphrase extraction and text classification techniques. We applied approaches from both tasks the existing and two new evaluation datasets and show that the performance of approaches is still quite low. Overall, we have shown that for most documents a TOC can be generated by detecting the hierarchical relations if the documents already contain segments with corresponding titles. In the other cases, one can use segment title generation, but additional research based on our newly created datasets will be necessary to further improve the task performance.

In future work, we want to develop a prototype of our search interface and perform user acceptance tests. Furthermore, we want to continue develop better features for the task of hierarchy identification, and want to create methods for post-processing a TOC in order to generate a coherent table-of-contents.

We made the newly created evaluation datasets and our experimental framework publicly available in order to foster future research in table-of-contents generation.[8]

Acknowledgements

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, by the Klaus Tschira Foundation under project No. 00.133.2008, and by the German Federal Ministry of Education and Research (BMBF) within the context of the Software Campus project *open window* under grant No. 01IS12054. The authors assume responsibility for the content. We thank Tri Duc Nghiem and Marko Martin for their valuable contributions and the we thank the anonymous reviewers for their helpful comments.

[8] Available at `http://www.ukp.tu-darmstadt.de/data/table-of-contents-generation/`

References

S.R.K. Branavan, P. Deshpande, and R. Barzilay. 2007. Generating a Table-of-Contents. In *Annual Meeting of Association for Computational Linguistics*, volume 45, pages 544–551.

T. Brants and A. Franz. 2006. Web 1T 5-gram Corpus version 1.1. Technical report, Google Inc., Philadelphia, USA.

J. Brooke and G. Hirst. 2011. Native Language Detection with 'cheap' Learner Corpora. In *Learner Corpus Research 2011 (LCR 2011)*.

C. Carpineto, S. Osiński, G. Romano, and D. Weiss. 2009. A Survey of Web Clustering Engines. *ACM Computing Surveys (CSUR)*, 41(3):17.

Z. Chen. 2011. Collaborative Ranking: A Case Study on Entity Linking. pages 771–781.

M. Collins and B. Roark. 2004. Incremental Parsing with the Perceptron Algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 111–118.

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. 2001. *Introduction to Algorithms*. The MIT press, Cambridge, MA, USA, 2nd edition.

A. Csomai and R. Mihalcea. 2006. Creating a Testbed for the Evaluation of Automatically Generated Back-of-the-book Indexes. In *Computational Linguistics and Intelligent Text Processing*, pages 429–440.

H. Daumé and D. Marcu. 2005. Learning as Search Optimization: Approximate Large Margin Methods for Structured Prediction. *Proceedings of the 22nd International Conference on Machine Learning*, (1):169–176.

R. Eckart de Castilho and I. Gurevych. 2011. A Lightweight Framework for Reproducible Parameter Sweeping in Information Retrieval. In *Proceedings of the 2011 workshop on Data infrastructures for supporting information retrieval evaluation*, DESIRE '11, pages 7–10, New York, NY, USA.

J. Feng, P. Haffner, and M. Gilbert. 2005. A Learning Approach to Discovering Web Page Semantic Structures. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1055–1059. Ieee.

J.R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

E. Frank, G.W. Paynter, and I.H. Witten. 1999. Domain-specific Keyphrase Extraction. In *Proceedings of 16th International Joint Conference on Artificial Intelligence*, pages 668–673.

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

R. Jin and A.G. Hauptmann. 2001. Automatic Title Generation for Spoken Broadcast News. In *Proceedings of the first international conference on Human language technology research*, pages 1–3. Association for Computational Linguistics.

C. Lopez, V. Prince, and M. Roche. 2011. Automatic Titling of Articles using Position and Statistical Information. *Proceedings of the International Conference on Recent Advances in Natural Language*, pages 727–732.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

G. Mikros and E.K. Argiri. 2007. Investigating Topic Influence in Authorship Attribution. In *Proceedings of the SIGIR 2007 International Work- shop on Plagiarism Analysis, Authorship Identifica- tion, and Near-Duplicate Detection, PAN 2007*.

L.M. Nguyen and A. Shimazu. 2009. A Semi-supervised Approach for Generating a Table-of-Contents. In *Proceedings of the International Conference RANLP-2009*, number 1, pages 312–317.

F.C. Pembe and T. Güngör. 2010. A Tree Learning Approach to Web Document Sectional Hierarchy Extraction. In *Proceedings of 2nd International Conference on Agents and Artificial Intelligence*.

H. Schmid. 1994. Probabilistic Part-of-Speech Tagging using Decision Trees. In *Proceedings of International Conference on new Methods in Language Processing*, volume 12, pages 44–49.

K. Stein and C. Hess. 2007. Does It Matter Who Contributes? - A Study on Featured Articles in the German Wikipedia. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 171–174, New York, NY, USA.

P.D. Turney. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2(4):303–336.