

Truncation: All the News that Fits We'll Print*

Joshua Hailpern
HP Labs
Palo Alto, CA, USA
joshua.hailpern@hp.com

Niranjan Damera Venkata
HP Labs
Chennai, Tamil Nadu, India
niranjan.damera-venkata@hp.com

Marina Danilevsky
University of Illinois
Urbana, IL, USA
danilev1@illinois.edu

ABSTRACT

A news article generally contains a high-level overview of the facts early on, followed by paragraphs of more detailed information. This structure allows copy editors to truncate the latter paragraphs of an article in order to satisfy space limitations without losing critical information. Existing approaches to this problem of automatic multi-article layout focus exclusively on maximizing content and aesthetics. However, no algorithm can determine how “good” a truncation point is based on the semantic content, or article readability. Yet, disregarding the semantic information within the article can lead to either overly aggressive cutting, thereby eliminating key content and potentially confusing the reader; conversely, it may set too generous of a truncation point, thus leaving in superfluous content and making automatic layout more difficult. This is one of the remaining challenges on the path from manual layouts to fully automated processes with high quality output. In this work, we present a new semantic-focused approach to rate the quality of a truncation point. We built models based on results from an extensive user study on over 700 news articles. Further results show that existing techniques over-cut content. We demonstrate the layout impact through a second evaluation that implements our models in the first layout approach that integrates both layout and semantic quality. The primary contribution of this work is the demonstration that semantic-based modeling is critical for high-quality automated document synthesis within a real-world context.

Categories and Subject Descriptors

I.7.2 [Document Preparation]: Desktop Publishing; I.7.4 [Document and Text Processing]: Electronic Publishing

Keywords

Truncation; Pagination; Novelty; Semantic; Cut; SLM

*The title is a play on words based on the New York Times tagline, “All the News that’s fit to print”

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
DocEng '14, September 16–19, 2014, Fort Collins, Colorado, USA.
Copyright 2014 ACM 978-1-4503-2949-1/14/09 ...\$15.00.
<http://dx.doi.org/10.1145/2644866.2644869>.

1. INTRODUCTION

Within the context of traditional newspapers or magazines, there are a finite amount of space within each issue limiting the amount of content being presented. A news article generally contains a high-level overview of the facts early on, followed by paragraphs of more detailed information [23, 24, 34]. This is done with two aims in mind: First, if the reader does not consume the entire text, “partial reading in that case will not result in partial understanding but only in missing a few, lower-level details.” [34]; Second, because the most relevant and important information is upfront, copy editors have the ability to cut paragraphs from the ends of the articles to make the included characters fit within space limitations without the loss of critical content [34]. Thus, copy editors and designers work to strike a balance between presenting enough detailed content for readability, while also accounting for space constraints and the aesthetic presentation of the page.

Automated document composition algorithms have been introduced to automate this largely manual workflow. Current approaches to automated document synthesis/layout focus extensively on content and aesthetic maximization [18, 19]. These optimization algorithms work with a series of rules to maximize the presentation within the spatial and visual constraints.

While these approaches have largely been successful at addressing issues of layout, they do not take into account the *semantic* content of the articles. This can result in either cutting too aggressively, resulting in missing key information and potentially misrepresenting the article content, or forcing superfluous content to remain potentially lowering the flexibility of layout algorithms.

Thus, the primary contribution of this work is an algorithm that predicts the likelihood that a given cut point is too short (missing key content) or too long (showing excessive details) based on article semantic content and readability. To the authors’ knowledge, this is the first such semantic-based truncation algorithm, allowing layout algorithms to prioritize discarding the least critical content, thus maximizing both layout, aesthetics, and readability.

In this paper we performed an extensive user study on over 700 news articles, in which participants rated 8 potential cut points as being too short or too long. Using all data collected, we extracted readability and semantic content features and used them to create two predictive models: one that determines the probability that a given truncation point is too short (and should have a high penalty if cut) and a second that predicts if a cut point is too long (reduc-

ing the value of including it in a final layout). These models were then tested within the context of an existing newspaper layout algorithm with promising results.

2. RELATED WORK

For journalists who write news articles, truncation of content by layout editors is expected. Thus, articles are structured so that it is possible to truncate and fit them within the confines of a print layout [23, 24, 34]. Though the style and exact structure varies by subject domain [23], journalists construct articles to be overly long so as to provide editors with as much flexibility as possible [34].

For automatic news layout systems, truncation is viewed as a byproduct of the layout system- given spatial constraints, articles are trimmed to reduce their length to maximize content (article count) and aesthetic presentation[1, 9, 3]. Subsequently, determining ideal truncation points beyond overall layout optimization (especially considering the semantic content therein) has not been explored in the related literature.

In this section we highlight three related problem spaces to Truncation from which we can inform our approach.

2.1 Pagination

Pagination, unlike truncation, preserves all of an article’s content while splitting it over multiple pages. Pagination approaches aim to create ‘high-quality documents,’ defined as documents ‘without unwanted empty areas’ [8]. Within the context of a single document, such as a book or a paper, pagination impacts the layout relationship of text content to related figures or tables [6].

However, the problem’s complexity increases for automated document layout of newspapers or magazines[13, 18, 19]. In addition to optimizing the placement of figures within text, a given page can also contain content from multiple articles, and all content must fit within a predetermined page count. When determining how to break up articles across multiple pages, the preferred approach is to use constraint-based layout models, in which layout specifications are described by linear constraints imposed on items [2, 20, 25, 31, 38].

2.2 Automatic Summarization

A similar problem to truncation is that of single document summarization - the process by which the text of an article is reduced either by extraction (lifting sentences from the original text)[26] or abstraction (using natural language processing techniques to generate new sentences)[12]. While these approaches are not used by layout or copy editors, they are relevant to the construction and evaluation of our semantic-based break point techniques.

Corpus summarization uses a large collection of documents to build a model of the topics being discussed (e.g. topic modeling [5, 7], SumBasic[14], KLSum[28]) or opinions rendered (e.g. Opinion Mining [16]). Corpus summarization approaches rely upon a large body of documents (e.g., a collection of tweets [7]) from which patterns about the ‘whole’ collection can be derived, and are generally more powerful because they have more data from which to draw summaries. In contrast, single article summarization [23, 22] utilizes only one article to create a summary. Within single article summarization, most algorithms are designed to summarize long (e.g. book)[37], well structured (e.g. chapters or sections) text[10, 32, 36], thus maximizing the amount of se-

mantic and structural cues from which to derive summaries. The most notable exceptions to single unstructured documents are TextRank[26] and LexRank[11]. Both of these algorithms use a simple graph-based approach, treating each sentence as a node. The summary sentence of the article is calculated by finding the centroid of the graph based on a distance vector. There has been some work specifically with summarizing news articles by extracting the most important facts from the article [21]. Finally, it has shown that using the first n sentences of a news article as a summary performs very well [27], which adds motivation for our problem.

2.3 Article Sub-Topic Segmentation

Another tangential, though separate problem space is that of article subtopic segmentation, most notably TextTiling [17, 29]. The main focus of subtopic segmentation is to divide a single article into subtopics, or sections. The goal is not to rank the sections on importance or quality, but rather to identify the conceptual shifts in the content. TextTiling is a straightforward and simplistic [4] method that examines lexical co-occurrences of terms between phrases in a document, and identifies sharp changes to be breaks between subtopics. However, the lack of ranking or quality of each subtopic shift means that such approaches’ direct application to the truncation problem are limited, and a direct comparison of TextTiling with this problem is difficult. However, this work does suggest broad techniques to build upon, such as detecting breaks based on semantic change variation across an article relative to a mean change, and using vocabulary occurrences to signify content themes [17].

3. SCOPING & KEY TERMS

The goal of this work is to create and test a novel algorithm that can use a news article’s semantic content and readability (rather than only layout/spacing optimization) to determine the probability a given cut point in an article is too short or too long for successful truncation. The challenge is twofold: first truncating an article too early risks eliminating key information and potentially confusing the reader as to the article’s content; and second, setting too generous of a truncation point leaves in superfluous content and makes automatic layout less flexible.

While there are many applications of this technique, this work specifically looks at news articles. Our novel approaches draw on and compare with an existing static truncation technique, and two semantic-based pagination techniques [15].

In this paper we first define our vocabulary and project scope. We then describe our article collection, followed by our truncation point evaluation experiment. Based on this data we present our modeling approaches to predict truncation point quality. Finally, we discuss results from applying our findings in a real-world context.

3.1 Key Terms

Topic: Refers to the use of the word in the English vernacular, such as the topic of a news article (e.g. a specific election in a specific country on a specific day), rather as it is used in the context of Topic Modeling approaches.

Article: A piece of written text about a specific topic (e.g., a specific New York Times article about an election in a country).

Cut Point: The location at which an article is ‘broken’, such that the content before the cut point is used, and the content after the cut point is never seen by the reader.

Minimal or Lower Bound Cut Point: The first (and thereby minimal) location within a news article which would make an acceptable cut point. In other words, the reader would have, by this time, understood enough about the topic being discussed, but has not yet seen all the details or nuances. Breaking before this lower bound is likely to cause the reader to miss a key concept or aspect of the article, and risk being misled about the content.

Truncation: The task of determining the cut point(s) of an article.

Corpus: A collection of articles on the same topic (e.g. 100 news articles about the same specific bombing event).

Subject: An overarching thematic grouping of articles or corpuses (e.g. Sports, Politics)

Document: A set of sentences (a single article, subset of an article, multiple articles), including related images and their layout on a page(s).

Semantics: Refers to the use of the word in the English vernacular, focusing on the *meaning* of words or phrases.

3.2 Pagination & Summarization

It should be noted that there is an important distinction between truncation (the focus of this paper) and pagination (not discussed in this paper - see [15]). Pagination, as used in this context, implies that after the break point, a reader will see more of the given article, usually on a subsequent page. On the contrary, when an article is truncated, the portion of the article which follows the break point will never be presented to the reader in any form. We do *not* deal with pagination tasks in this work.

Further, this work does not address the task of document summarization or subtopic identification. Within the context of automatic news article layouts, article content is not summarized, and the content is preserved in the order and sentence structures of the journalist. Furthermore, subtopic breaks do not rank or determine which points are of more or less importance. While we do cite relevant summarization, subtopic identification and IR literature, we use these approaches to help inform our feature selection and readability metrics - but they are not directly applicable to solving the task of truncation quality prediction.

4. EXPERIMENT: DATA COLLECTION

The primary goal of this work is to develop and test models that can predict if any given cut point would be missing key content (too short) or allows for too large an amount of superfluous content (too long). To this end, we conducted a data collection experiment on Amazon Mechanical Turk (MT) which generated a hand coded dataset of articles with cut point quality ratings. This section details our experimental methods for data collection.

Subject	Total # Corpuses	Total # Articles	μ Articles per Corpus (sd)
US	183	19429	106.17 (52.81)
Business	166	19209	115.72 (63.45)
Politics	129	25094	194.53 (122.53)
Ent.	218	30534	140.06 (78.70)
World	158	20440	129.37 (75.59)
Sports	203	27463	135.29 (65.08)
Tech	232	42104	181.48 (119.72)
All	1289	184273	142.96 (90.73)

Table 1: Corpus Summary Statistics
Data Collected from Google News from 10/13 to 11/13

4.1 Hypotheses

The data collected from this experiment will be used to address four research hypotheses:

- H1 A model can be constructed that can predict the likelihood that a truncation cut point is missing key content (too short)
- H2 A model can be constructed that can predict the likelihood that a truncation cut point has superfluous content (too long)
- H3 A model can be constructed that can determine the minimal truncation point (not too long or too short)
- H4 The identification of a good truncation point for an article is functionally different than that of a good pagination point.

All these hypothesis are from the perspective of the reader of a news article (not necessarily a layout editor, copy editor, or journalist).

4.2 Article Data Set

We collected corpuses of news articles by scraping Google News every 8 hours, from October 21 to November 27 2013¹. Corpuses were ‘accepted’ if they contained at least 50 unique articles (as aggregated by Google News). The source HTML of the original articles from the originally hosted websites were retrieved, allowing the body copy to be extracted using CETR [35]. Because “... paradigmatic discourse structure differs significantly over text genres and subject domains” [23], we collected corpuses from 7 subject areas: World News, US News, Entertainment, Business, Technology, Sports and Politics. Descriptive statistics of the downloaded corpuses can be found in Table 1.

To conduct this experiment, we randomly selected 1 article from 100 randomly selected corpuses from each of 7 subject areas. This ensures a wide variety of topics being discussed. The resulting data set consisted of 700 articles. Descriptive and readability statistics for the randomly selected 700 articles are presented in Table 2.

4.3 Mechanical Turk Methods

MT HITs were constructed from the 700 articles. A HIT is an individual task given to a person on MT. HITs were not grouped together (containing more than one article) and presentation order was random (reducing order effects). Each HIT consisted of a brief definition of a cut point, minimal truncation point, the source text and the 8 candidate cut

¹Collection stopped when at least 100 corpuses per subject area were collected.

	Grade Level	Reading Level	Fog Index	Sentence Count	Word Count
Business	11.35 (3.53)	47.35 (11.29)	7.63 (3.45)	56.70 (25.50)	911.23 (180.56)
Ent	13.56 (4.26)	47.92 (12.91)	11.29 (4.30)	36.79 (15.31)	921.50 (217.30)
Politics	13.87 (3.46)	41.75 (10.71)	10.42 (3.52)	37.94 (12.75)	888.33 (160.91)
Sports	10.51 (2.47)	59.29 (8.63)	8.93 (2.43)	40.37 (12.61)	839.67 (164.86)
Tech	12.04 (3.07)	51.15 (11.33)	9.58 (2.82)	41.20 (12.70)	923.77 (225.70)
US	13.06 (3.11)	45.86 (11.14)	10.04 (3.03)	37.72 (10.70)	881.95 (171.30)
World	14.37 (3.61)	39.06 (12.59)	10.63 (3.46)	36.89 (11.67)	896.77 (182.52)
All	12.68 (3.64)	47.48 (12.84)	9.79 (3.52)	41.08 (16.55)	894.78 (189.57)

Table 2: Article Source Statistics
100 Articles for each Subject (700 overall) | Mean and Std Values Reported

points presented in random order (reducing order effects). Before each sentence in the source text, we inserted the characters ‘(SX)’ where X was the sentence number. We used these indices to refer to a given cut point. For each cut point, the participants were asked to respond to the statement *Is (SX) a good Cut Point?* using a 7-point scale (*Too Short - Missing Content* (1) to *Minimally Acceptable Cut Point* (4) to *Too Long - Extra Details* (7)). Each HIT was completed by 2 Master level Turkers², yielding 5600 measures.

To ensure ‘legitimate’ HIT completion, one ‘sanity check question’ was included asking Turkers to find the N th word in the M th sentence. In addition, a HIT was rejected if the Turkers’ response failed basic logic checks on their responses. First, all responses could not be the same (e.g. not all 6s). Second, if a single HIT asked about the same cut point more than once³, that individual Turker’s response must be the same to both questions (e.g. rating sentence #4 as a 6 then rating sentence #4 as a 2). Third, ratings must be in chronological order. For example, labeling a sentence early in the article as “too long”, and then a subsequent sentence as “too short”, does not make sense. However succeeding break points (in article order) could have the same ranking. If any of the three logic tests or the sanity check were failed, the HIT was rejected and re-posted. Participants were remunerated 30 cents per accepted HIT.

4.4 Eight Candidate Cut Points

Our eight candidate cut points were selected as follows:

Truncation Baseline – One of the candidate cut points was set at the first 20% of an article (measured by character count), then rounded up to the end of the current paragraph. This approach (the modus operandi for constructing printable newspapers at Company XYZ) assumes that within the first fifth of an article, key concepts have been presented to the reader, and are roughly bounded by paragraph delimitation. We refer to this technique as **Twenty Percent**.

Pagination Baseline – As discussed in our related work, the only semantic based breaking technique was [15], though the focus was on pagination not truncation. We therefore explicitly repurpose their two minimal pagination break point algorithms to answer H4. We will refer to their Novelty Article algorithm as **Novelty**, and their SLM Article algorithm as **SLM**. However, unlike [15], we take a more conservative approach, and will round the Novelty and SLM points up to the end of the current paragraph for any given cut point to reflect the bias that human judgement is more likely to favor a cut point located at the end of a paragraph.

²Approval Rate above 95% and at least 1000 approved HITs

³The three baselines could produce the same cut point

Random Sample – The remaining 5 candidate points were randomly selected from the remainder of the article. All random cut points are selected between paragraphs (preserving full paragraphs). If a cut point was already selected by a baseline, we re-select an alternative random cut point.

4.4.1 Method Limitations

Any evaluation with multiple comparisons has an interaction effect, in that the rating or quality of one cut point can be impacted by the other cut points offered for comparison. Thus, a cut point that might have been rated a 5 (slightly too long) when viewed in isolation, may be pushed ‘higher’ if there are other, less optimal cut points located earlier in the article. Thus, these results must necessarily be viewed in the broader context of the other cut points presented.

5. RESULTS: DATA COLLECTION

In this section, we describe the results of our data collection experiment. Summary statistics for each response rating are presented across all subjects in Table 3 Based on an examination of these hand-coded responses, we can draw conclusions regarding some of our hypothesis as well as inform how we will predict truncation point quality.

5.1 H3: No “ideal” Truncation Point

H3 hypothesized that a model could be constructed to predict a minimally acceptable truncation point in an article. However, when we examine Table 3, a very small percentage of all ratings were a 4 (minimally acceptable truncation point). Moreover, this implies that the granularity tested in this experiment (7-point) was too fine. Subsequently, predicting only one rating (be it 4 or any of the others) would rely upon too small a data set to build an accurate data set and not end up modeling the “noise.”

Based on this observation (that H3 should be rejected) our subsequent experiment to predict the truncation point quality collapses all points with rating under 4 as “too short,” and all ratings above 4 as “too long.” This will focus on broader trends in the data.

5.2 Twenty Percent is Too Short

Of the eight candidate cut points, one was Twenty Percent, a “reasonable” lower bound for news article layout. The results for this subset of truncation points is presented in Table 4. Surprisingly, Twenty Percent appears to cut an article’s content overly aggressively, with the median value never reaching a score of 3 (which would still be too short). However, when we consider that 20% is a relative value based entirely on the length of the article, it is reasonable to con-

	User Response		Sentence Number			KLD Score			NVL Score			Percent of Article		
	Rank	%	Mean (sd)	Median	Hist	Mean (sd)	Median	Hist	Mean (sd)	Median	Hist	Mean (sd)	Median	Hist
All Subjects	1	17.43	7.47 (4.01)	7.00		0.37 (0.15)	0.35		0.31 (0.17)	0.29		0.16 (0.09)	0.15	
	2	20.45	14.02 (7.05)	12.00		0.28 (0.13)	0.26		0.46 (0.20)	0.44		0.31 (0.14)	0.28	
	3	16.98	22.36 (12.26)	19.00		0.20 (0.14)	0.18		0.62 (0.24)	0.62		0.49 (0.25)	0.43	
	4	9.12	21.28 (12.24)	17.00		0.21 (0.15)	0.20		0.61 (0.24)	0.57		0.48 (0.26)	0.39	
	5	11.39	24.27 (10.31)	22.00		0.18 (0.13)	0.16		0.67 (0.22)	0.66		0.55 (0.22)	0.51	
	6	10.06	29.94 (7.74)	29.00		0.13 (0.10)	0.10		0.77 (0.18)	0.79		0.67 (0.17)	0.68	
	7	14.58	40.38 (9.41)	39.00		0.06 (0.08)	0.04		0.89 (0.13)	0.95		0.85 (0.14)	0.88	

Table 3: Summary Statistics for each User Ranking Response

Ratings based on a 7-Point scale:

(1) “missing information” (4) “balanced” and (7) “excess text.”

	Algorithm	Mean (sd)	Median	Histogram
Overall	Twenty Percent	2.23 (1.11)	2.00	
	Novelty	2.90 (1.29)	3.00	
	SLM	2.58 (1.83)	2.00	
BUS.	Twenty Percent	2.12 (1.08)	2.00	
	Novelty	2.78 (1.22)	2.00	
	SLM	2.24 (1.52)	2.00	
ENT.	Twenty Percent	2.82 (1.37)	2.50	
	Novelty	3.30 (1.52)	3.00	
	SLM	3.28 (2.04)	3.00	
POL.	Twenty Percent	2.27 (1.02)	2.00	
	Novelty	2.92 (1.19)	3.00	
	SLM	2.89 (1.98)	2.00	
SPT.	Twenty Percent	1.96 (0.95)	2.00	
	Novelty	2.60 (1.21)	2.00	
	SLM	1.99 (1.50)	1.00	
TECH.	Twenty Percent	2.21 (1.25)	2.00	
	Novelty	2.90 (1.45)	2.50	
	SLM	2.49 (1.71)	2.00	
US	Twenty Percent	2.16 (0.87)	2.00	
	Novelty	2.99 (1.15)	3.00	
	SLM	2.74 (1.92)	2.00	
WLD	Twenty Percent	2.06 (0.98)	2.00	
	Novelty	2.81 (1.15)	3.00	
	SLM	2.44 (1.81)	2.00	

Table 4: Lower Bound Prediction Performance

In Histogram, ratings of 4 are in black

clude that an article’s key content is not based exclusively on a relative percentage of the document. This is backed up by the data in the last column of Table 3; When we examine the percent of an article each user rank is at, there is a large standard deviation. This makes the raw document percentage (be it 20% or any other percentage) an unlikely lone predictor of minimal truncation point. That said, percentage may be useful when combined with other features for predicting quality of a truncation point.

5.3 H4: Truncation is not Pagination

While pagination is conceptually different from truncation, there are parallels between the two techniques. In H4 we hypothesized that the cut point at which readers would feel comfortable paginating an article would be different than where an article would be truncated. To test this hypothesis, semantic-based minimal pagination point algorithms [15] to generate two of the eight candidate break points, since their mean performance in a pagination experiment were rated 4.66 and 5.83 (SLM and Novelty respectively). However, readers considered these points to be too short for the truncation task, as seen in Table 4.

Upon reflection, these results make conceptual sense. Readers are more conservative when they know that after the cut, there will be no more content. Therefore, they wish to en-

sure a higher level of detail in the presented article. This confirms H4.

6. EXPERIMENT: MODELING

In this section, we detail the features extracted for each cut point, modeling techniques tested, and the methods used to evaluate model performance. To simplify our discussion of features, we divide them into Readability Measures (features that describe the article as a whole), and Incremental Measures (features that describe the current truncation point and change throughout the article). The subsequent section will then highlight the resulting performance results of the models.

6.1 Measures: Readability

For each candidate cut point, we associate a set of features based on the readability of the *entire* article. This allows the modeling algorithms to account for the complexity, length, and quality of the article as a whole. Grade Level was calculated using the Flesch-Kincaid grade level, which indicates that a student at that current U.S. school grade should be able to understand said article (7.0 to 8.0 is ‘optimal.’) Reading Level, or Flesch Reading Ease rates text on a 100 point scale, with higher scores meaning easier to understand (60-70 is ‘optimal.’) The Fog Index is another measure that indicates the years of education to understand a article in a single reading (8.0 is ‘optimal.’) Overall sentence count and word count were included as additional features.

It should also be noted that the subject area of the article was also used as a feature, allowing the algorithm to account for changes in writing styles of each article’s subject area.

6.2 Measures: Incremental

The first feature we developed, **NVL** is predicated on the idea that as a reader traverses an article, he or she is exposed to key concepts/words. The initial exposure to a given key word is enough to make the reader aware that the subject is relevant to the article⁴. To this end, the Keyword Novelty approach attempts to find when the reader has been exposed to ‘enough’ of these key words so that he or she would have a general understanding of the article.

The second feature, **KLD**, is based on the idea that there is a probabilistic distribution of words (and their frequencies) that are ‘ideal,’ and we should strive to mimic (e.g. the full text of the article). We use a Statistical Language Model approach (SLM) to create a model of the ‘full’ article, and, for a given portion of an article being made visible to the reader, we use an information theoretic measure (KL Divergence) to evaluate how closely the model of that ar-

⁴Exactly how the given subject is related to the full text may not be clear until later in the article.

ticle portion matches the ‘ideal’ model of the entire article (implying that the article portion conveys much of the same content). The value of the KL Divergence metric at every sentence can be used as a feature when constructing our models. An added benefit of the SLM approach is the ability to smooth the keyword frequencies that are common to the broad subject (in this work we use Dirichlet Prior Smoothing, which has been shown to be effective [39]).

In addition to Novelty Score and KL Divergence, we also included non-semantic features such as the number of completed sentences at the current cut point (**SEN**) and the percent of the article (based on word count) at the cut point (**PRC**). While not as complex as NVL of KLD, these measures are seen in much of the related work and therefore may be useful features for truncation point quality prediction.

6.2.1 Calculating Keyword Novelty

The first step is determining the article keywords. Following standard IR techniques, we limit an article’s text to information-heavy words (nouns), and remove any pluralization through lemmatization. However, not all remaining keywords are equally relevant to the article in question. Commonly, term frequency (TF) can be used as a proxy for keyword relevance. However, TF is generally not robust on short or sparse data, such as a single newspaper article.

An alternative technique to discovering keywords is to use Singular Value Decomposition (SVD). SVD is able to filter out the noisy aspects of relatively small or sparse data, and is often used for dimensionality reduction. To repurpose SVD to calculate word weight, we represent each sentence as a row in a sentence-word occurrence matrix encompassing m sentences and n unique words, which we will refer to as $\mathbf{M}$ (which can be constructed in $O(m)$).

SVD decomposes the $m \times n$ matrix $\mathbf{M}$ into a product of three matrices: $\mathbf{M} = \mathbf{U} \Sigma \mathbf{V}^*$. Σ is a diagonal matrix whose values on the diagonal, referred to as σ_i , are the singular values of $\mathbf{M}$. By identifying the four largest σ_i values, which we refer to as $\lambda_1 - \lambda_4$, we are able to take the corresponding top eigenvector columns of $\mathbf{V}$ (which is the conjugate transpose of $\mathbf{V}^*$), which we refer to as $\xi_1 - \xi_4$. Each entry in each of these vectors $\xi_1 - \xi_4$ corresponds to a unique word in $\mathbf{M}$.

We then create ξ' , a master eigenvector calculated as the weighted average of $\xi_1 - \xi_4$, weighted by $\lambda_1 - \lambda_4$:

$$\xi' = \frac{1}{4} \sum_{i=1}^4 \lambda_i \xi_i \quad (1)$$

Thus, ξ' is a vector in which each entry represents a unique word, and the value can be interpreted as the ‘centrality’ of the word to the given article [37].

Once we have the keyword weights,⁵ we iterate over each sentence in a given article. The ‘value’ of a sentence is the sum of all of the unique keywords’ weights seen up to (and including) that sentence. Thus, the weight of a given sentence is a cumulative sum and each keyword only contributes to the overall sum on its first occurrence.

6.2.2 Calculating KL Divergence

As in Keyword Novelty, we pre-filter the text in each article to only contain the lemmatized high-information (noun) words.

⁵To reduce noise, we use the top 500 words from SVD [30].

In order to explain our use of SLM, consider $\mathbf{S}$ to be the set of all subjects ($s_0 \dots s_g$) in our dataset. We work with one subject at a time, which we will refer to as s_i .

Let $\mathbf{D}$ be the set of all articles ($d_0 \dots d_b$) in the given subject s_i and $\mathbf{W}$ to be the set of all unique words ($w_0 \dots w_h$) in s_i .

Denote the frequency of a given word w_j in a given article d_k as $f(w_j|d_k)$. Then the total count of all words in d_k is calculated as:

$$T(d_k) = \sum_{j=0}^h f(w_j|d_k) \quad (2)$$

and the probability of a given word (w_j) in d_k is:

$$p(w_j|d_k) = \frac{f(w_j|d_k)}{T(d_k)} \quad (3)$$

This therefore allows us to calculate the probability of a word in an article, using Dirichlet Prior smoothing [39]:

$$q(w_j|d_k) = \frac{f(w_j|d_k) + \mu * p(w_j|s_i)}{T(d_k) + \mu} \quad (4)$$

where $p(w_j|s_i)$ is the occurrence probability of the word w_j in the entire subject s_i :

$$p(w_j|s_i) = \frac{\sum_{d_k \in s_i} f(w_j|d_k)}{\sum_{d_k \in s_i} T(d_k)} \quad (5)$$

and where the smoothing constant μ is estimated using [33]:

$$\begin{aligned} m_j &= p(w_j|s_i) \\ B_j &= \sum_{d_k \in s_i} ((\frac{f(w_j|d_k)}{T(d_k)} - m_j)^2) \\ \mu &= \frac{\sum_{w_j \in W} \frac{B_j}{m_j * (1 - m_j)}}{\sum_{w_j \in W} \frac{B_j^2}{m_j^2 * (1 - m_j)^2}} \end{aligned} \quad (6)$$

As we traverse an article sentence by sentence, we redefine the variable $\mathbf{N}$ to refer to the subset of sentences in a given article we have seen. Thus, we calculate the probability of a word in $\mathbf{N}$, using Dirichlet Prior smoothing [39] as:

$$q(w_j|N) = \frac{f(w_j|N) + \mu * p(w_j|s_i)}{T(N) + \mu} \quad (7)$$

To compare each successive test SLM to the Ideal article, we use the KL-Divergence metric (smaller is better):

$$KLDivergence = \sum_{w_j \in d_k} (\ln(\frac{q(w_j|d_k)}{q(w_j|N)})) * q(w_j|d_k) \quad (8)$$

Thus for each sentence or paragraph in an article, we can calculate the KLDivergence score and use it as a feature.

6.3 Modeling Techniques

In order to predict the probability a truncation cut point was “too short” or “too long,” we construct a series of predictive models using Weka:

- Logistic regression (LGST) - used extensively to predict the probability of an event’s occurrence
- J48 decision tree (TREE) - a classifier which uses the concept of information gain to build a decision tree from data features
- Random Forest (RFST) - an ensemble classifier, combining multiple concurrently built decision trees

Rank	Classifier	Measures	TP	TN	FP	FN	Kappa	aroc	Accuracy	Inaccuracy	Sens.	Spec.	Prec.	F1
1	TREE	NVL PRC SEN	4104	4624	1529	959	0.557	0.808	0.778	0.222	0.811	0.752	0.729	0.767
2	TREE	PRC SEN	4056	4636	1517	1007	0.550	0.803	0.775	0.225	0.801	0.753	0.728	0.763
3	TREE	KLD SEN	4053	4644	1509	1010	0.550	0.822	0.775	0.225	0.801	0.755	0.729	0.763
4	TREE	KLD NVL SEN	4049	4654	1499	1014	0.551	0.827	0.776	0.224	0.800	0.756	0.730	0.763
5	TREE	KLD PRC SEN	4047	4639	1514	1016	0.548	0.803	0.774	0.226	0.799	0.754	0.728	0.762
14	LGST	KLD NVL SEN	3601	5096	1057	1462	0.543	0.853	0.775	0.225	0.711	0.828	0.773	0.741
32	RFST	KLD SEN	3182	4253	1900	1881	0.320	0.793	0.663	0.337	0.628	0.691	0.626	0.627
46	SVM	KLD NVL PRC SEN	2973	4405	1748	2090	0.305	0.652	0.658	0.342	0.587	0.716	0.630	0.608

Table 5: Less Than 4 “Too Short” Top Five Models (Across All Subjects)

Sorted by Sensitivity | For any classification technique not in the top 5, first occurrence and rank is shown

Rank	Classifier	Measures	TP	TN	FP	FN	Kappa	aroc	Accuracy	Inaccuracy	Sens.	Spec.	Prec.	F1
1	LGST	KLD PRC	6215	2720	1320	961	0.550	0.879	0.797	0.203	0.866	0.673	0.825	0.845
2	LGST	KLD PRC SEN	6221	2711	1329	955	0.549	0.880	0.796	0.204	0.867	0.671	0.824	0.845
3	LGST	NVL PRC SEN	6215	2719	1321	961	0.550	0.878	0.797	0.203	0.866	0.673	0.825	0.845
4	LGST	PRC SEN	6220	2711	1329	956	0.549	0.879	0.796	0.204	0.867	0.671	0.824	0.845
5	LGST	KLD NVL PRC	6207	2728	1312	969	0.550	0.879	0.797	0.203	0.865	0.675	0.826	0.845
13	TREE	NVL SEN	5784	3195	845	1392	0.580	0.844	0.801	0.199	0.806	0.791	0.873	0.838
31	SVM	KLD NVL PRC SEN	5766	2050	1990	1410	0.321	0.655	0.697	0.303	0.804	0.507	0.743	0.772
39	RFST	KLD	5513	2363	1677	1663	0.353	0.805	0.702	0.298	0.768	0.585	0.767	0.768

Table 6: Greater Than 4 “Too Long” Top Five Models (Across All Subjects)

Sorted by F1 | For any classification technique not in the top 5, first occurrence and rank is shown

- Support Vector Machines (SVM) - supervised learning model that constructs a classifier

To ensure robustness, we performed a 5-way cross validation (training a model on a random 80% of the data points and holding 20% in reserve for accuracy testing).

Each model was built using all readability measures. However, having more features can also mean extra noise and a reduction in quality. We therefore created multiple models using different permutations of our 4 Incremental Features. This allowed us to easily uncover which features are useful, and which are harmful or superfluous (some of the modeling techniques like Decision Tree or Random Forest prune features that are not used as decision points, thus automatically reducing the feature space.)

6.4 Modeling Evaluation

There are numerous evaluation metrics for model quality, from sensitivity to kappa. However, choosing the appropriate metric for the specific context of the problem is critical for maximizing the models’ performance.

In this regard, for the “too short” model, we chose to maximize **Sensitivity**, the measure that focuses on the true positive rate. In short, by not labeling a truncation point that is too short, critical article content is missed. Thus, we need to ensure that the actual positives are correctly identified.

The “too long” model, however, has a less detrimental outcome if there are false positives or false negatives. For this model, overall performance is key, and thus we used the **F1 Score**, the harmonic mean of precision and sensitivity (thus providing a more holistic performance assessment)⁶.

7. RESULTS: MODELING

Results from the various “too short” and “too long” modeling techniques are presented in Tables 5 and 6 respectively. While these models were trained on data from all 7 subjects, they take into account the subject area when predicting the quality of the truncation point⁷.

⁶In many situations, accuracy is used as a holistic measure. However, with sparse data, accuracy is a poor measure of performance because always guessing 0, or “not part of classification” can lead to high accuracy but poor performance.

⁷We did create models for each individual subject area, however,

Random Forest and SVM both performed surprisingly poorly. Further, the performance for each outcome was largely stratified based on classification technique, rather than features used. Subsequently, the top “too short” models were all variations of J48 Tree, and the top “too long” models were Logistic Regression⁸.

Using the J48 Tree and Logistic Regression, we can extrapolate a scalar value that indicates the probability that a given truncation point is either “too short” or “too long.” These values can then be used to inform the layout of multiple articles within a document and determining where to cut. For logistic regression, we apply the output of the model to a sigmoid function to get a probabilistic outcome (probability that the given point is “too long”).

For the decision tree, each leaf contains both the outcome and the number of instances of the training data assigned to that leaf, and how many of those instances were incorrectly classified. Thus, we can calculate the probability that a truncation point that reaches a given leaf is “too short” by the following equations: $Not\ Too\ Short = 0 + \frac{incorrectly\ classified}{total\ assigned}$ and $Too\ Short = 1 - \frac{incorrectly\ classified}{total\ assigned}$

The strengths of these two models, as shown in Tables 5 and 6, confirm our hypotheses H1 and H2, showing that we can create predictive models that can determine if a truncation point is too short or too long, based on reader impression. These results indicate that these models generate more complete (from a readability perspective) truncated news articles than existing non-semantic techniques.

8. LAYOUT IMPACTS

We now illustrate the utility and application of the content based article truncation algorithms in producing practical document layouts. Specifically, we examine the impact of article truncations on layout quality. As discussed earlier, layout designers strike a balance between what semantic content to keep and how to present it. In general, layout quality focuses on making documents to be more aesthetically pleasing by creating room for images or by ensuring

these models were nearly identical to subsets of the global models which took subject area into account. We therefore simplified our entire process by using the global models.

⁸When building models for each subject area, these top modeling techniques were consistently the best performing.

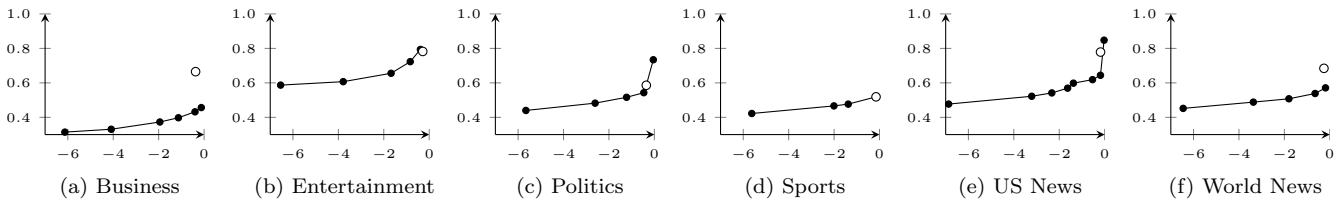


Figure 1: Semantic Content vs. Presentation Tradeoffs

- Semantic based plots are solid circles | — Semantic efficiency frontier is solid line | ◦ Non-Semantic layout is empty circle
- Y-Axis represent "Probability too Long" | X-Axis represents Layout Quality



(a) Non-Semantic Truncation

(b) Semantic Truncation

Figure 2: Examples of Article Layouts

All article text and images used with permission of the Associated Press

that blank space on a page is filled. This section, to the authors knowledge, is the first automated parallel to the traditional human drive approach that maximizes both layout and semantic content.

8.1 News Articles & Layout

To illustrate the impact of semantic based layout, we created six different layouts of news content, each taken from one of six content subject areas (Business, Sports, Entertainment, World news, US news and Politics). Articles, along with any corresponding images, were collected from the AP news feed.

Each subject area layout consisted of three long main articles and four short sidebar articles are arranged across two-pages. This allows us to test both "too short" (sidebar articles) and "too long" (main articles) measures.

8.2 Measuring Layout Quality

In order to explore the automatic layout improvements of our truncation algorithm, we situated this experiment within the context of the probabilistic document model (PDM) [9, 1] as a measure of document layout quality. PDM explicitly models the dependency between key design choices⁹ including what content should be placed on each page, choice of relative arrangements for page elements (what template to use for each page), and template parameters (physical dimensions of page elements like images, whitespace etc.).

⁹The coupling between these design variables is explicitly modeled as a Bayesian network.

PDM is a *micro model* for document quality that associates a probability (or log probability quality score) distribution with each conditional design choice made on each page. These probability distributions (e.g. whitespace, template, variance of acceptable whitespace) are specified at design time by a document designer (see [9, 1]). As an example, PDM will give higher probability to the documents that use designer-preferred templates (to enhance use of images) and designer-preferred physical dimensions for page elements (e.g., image and whitespace deviating from preferred "aesthetic" values will be penalized). We use the layout synthesis method described in [1] to generate and score layouts.

8.3 Measuring Content: Without Semantics

As a baseline, we create layouts without semantic consideration. A reasonable approach makes the assumption that, since articles are written to be cut from the end [34], longer articles have more superfluous information and can be truncated more aggressively. We iteratively truncate paragraphs from the longest article (given all cuts made up to this point). We then evaluate each possible truncation, and choose the final truncation corresponding to the best layout score achieved. Thus, finding an ideal layout becomes a process of maximizing layout quality, and content quality is only implicitly included in the determination of which content to cut.

8.4 Measuring Content: Semantic Truncation

In order to account for the semantic information in each article, our two semantic truncation algorithms are re-calculated

for each paragraph, such that, if the article is truncated at the start of that paragraph, the *probability too short* measure gives the probability that the remaining content in an article will be missing key content, and the *probability too long* measure gives the probability that the remaining content in an article will contain superfluous content. Thus, these two measures of semantic quality can be used to compare the content included in a given layout. Given that multi-article layouts combine multiple articles, we compute a content truncation score for all included articles by averaging the corresponding “too short” and “too long” scores.

8.4.1 Content vs. Presentation Tradeoffs

Subsequently, any given document is measured by both the layout quality and the content quality. However, to construct the ideal document, we must choose from several possible documents corresponding to various truncation points. For each “truncation” opportunity, we can cut a paragraph from any of the N articles. To limit the number of possible documents to choose from, we set an upper limit on the number of paragraphs to cut, m . This results in N^m permutations of possible total truncation points. However, this simplistic approach results in an intractable number of permutations. Even a moderate set of possible cuts and a small number of articles yields a huge number of documents (e.g., 10 cuts and 3 articles yields $3^{10}=59,049$ documents).

Instead of using all possible permutations, we use a sub-optimal but reasonable approach of ordering all paragraphs from all articles in ascending order of “probability too short.” This allows us pick from articles that are the least likely to be missing key content. Then, among paragraphs that have the same “probability too short,” we sort in descending order according to the “probability too long” measure. Thus articles that are unlikely to be missing key content, and highly likely to have superfluous content, bubble to the top. This creates a well defined sequence of possible truncations that sweeps the content space.

Each possible truncation layout is mapped to a point in 3-D space corresponding to the three measures (2 semantic-based measures described above, and 1 for layout measure). With respect to the layout size, we create two 2-D plots: long main articles (prob. too long vs. layout quality) and short sidebar articles (prob. too short vs. layout quality). In each plot we want to minimize the probability an article is too short/too long while maximizing the layout quality. We can easily eliminate *inefficient* document permutations, those that have both a larger “too short/too long” score and also at the same time a lower layout quality score. The remaining points represent a legitimate *efficient* tradeoff between content and presentation. We refer to the set of document points as a pseudo-efficient frontier.¹⁰ The exact truncation to use may be chosen by selecting a frontier point above a layout quality threshold that minimizes the probability too long/short. This gives a natural tradeoff between content and presentation, thus automating this selection.

8.5 Results & Discussion

For each of the six subject areas, we extract and plot the frontiers in Figure 1. For the long main articles, the frontiers are curves. By sliding along the frontier, content semantics can be easily traded for presentation quality. Thus, an ideal document layout can be determined.

¹⁰Not all possible permutations are explored

8.5.1 Comparing Semantic & Non-Semantic Layout

For three areas (Business, World, US News) the non-semantic truncation created sub-optimal and *inefficient* layouts¹¹ showing that the content based approach yields better layouts and less superfluous content in these cases. For example, Figure 2 shows visual comparisons for the Business news case. Notice¹² that the non-semantic truncation includes superfluous content and a twitter link at the end of the Macy’s article that is unrelated to the content. In fact the semantic based truncation made space for additional images, increasing layout quality while also minimizing the probability “too long”.

For two of the area (Politics, Sports) the non-semantic truncation corresponds exactly to a semantic-based frontier point. For only one area (Entertainment) the non-semantic based truncation outperforms (albeit slightly) the semantic-based method. This non-semantic layout, however, was never considered by the frontier because we cannot explore all possible truncations to get a true *efficient frontier*. However, since we sweep the space in a principled manner, we are never too far from a possible optimal point.

In general the results show that the content-based truncation strategy can automatically harmonize content and presentation, something a skilled content editor, together with a publication designer, had to manually do in the past.

8.5.2 SideBar: Extremely Aggressive Truncation

Given the extremely aggressive truncation required to fit 4 short articles into the sidebars, all truncations that fit the content with reasonable quality had the same probability that an article was too short. In other words, our algorithm was quite sure (no variation) that the requisite truncation of the articles were way too short. Thus the frontier was a single point, representing the highest quality layout of the sidebar articles with this aggressive truncation.

This is a byproduct of the users that generated the ratings of truncation point quality. These reviewers were members of the general public, and subsequently were extremely conservative with their minimal acceptable content. This is in stark contrast to layout editors who are more comfortable aggressively cutting content to fit extremely small space constraints. New models could be generated if layout editors were tasked to rate truncation points, or members of the general public were explicitly told that articles needed to be truncated to fit within an extremely small area. We strongly suspect that minimal acceptable truncation points would be lower, and provide discrimination among the various very short truncation points.

9. FUTURE WORK

One potential direction of future work is to expand this approach to model expert layout editors and journalists as to their perceived truncation point quality. Given that this work focused exclusively on *readers’* perceived quality, editors and journalists may have a more aggressive or nuanced view as to what is superfluous or necessary.

10. CONCLUSION

Creating ideal layouts is a challenge for both human layout editors, and computer systems. Efficient solutions strike

¹¹Figure 1, non-semantic point is above and to the left of frontier

¹²PDF is high resolution, and can be zoomed in.

a balance between aesthetics and semantic content within articles. Prior to this work, computer layout systems truncated content from articles to create documents based almost exclusively on optimizing aesthetics, at the detriment of any semantic information. In response, we have created the first model of truncation quality (based on the content of the article) and the first layout approach that integrates both layout quality *and* article content quality. These models were built from data collected during an extensive user study, and the resulting document layouts were validated and compared to that of non-semantic based approaches (in a real world context). The semantic+layout quality documents utilized more images while maximizing the articles' semantic information.

11. REFERENCES

- [1] I. Ahmadullin and N. Damera-Venkata. Hierarchical probabilistic model for news composition. In *DocEng*, page 141, New York, New York, USA, Sept. 2013. ACM Request Permissions.
- [2] G. J. Badros, A. Borning, and P. J. Stuckey. The Cassowary linear arithmetic constraint solving algorithm. *TOCHI*, 8(4 (Dec)):267–306, Dec. 2001.
- [3] J. Batista and D. Oliverira. Two algorithms for automatic document page layout. In *DocEng*, page 141, New York, New York, USA, Sept. 2008. ACM Request Permissions.
- [4] D. Beeferman, A. Berger, and J. Lafferty. Statistical Models for Text Segmentation. *Machine learning*, 34(1-3):177–210, 1999.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.
- [6] A. Brüggemann-Klein, R. Klein, and S. Wohlfeil. On the pagination of complex documents. *Lecture Notes in Computer Science*, 2598:49–68, 2003.
- [7] F. Chua and S. Asur. Automatic Summarization of Events From Social Media. In *ICWSM*, 2013.
- [8] P. Ciancarini, A. Di Iorio, L. Furini, and F. Vitali. High-quality pagination for publishing. *Software—Practice & Experience*, 42(6), June 2012.
- [9] N. Damera-Venkata, J. Bento, and E. O'Brien-Strain. Probabilistic document model for automated document composition. In *DocEng*, page 3, New York, New York, USA, Sept. 2011. ACM Request Permissions.
- [10] H. P. Edmundson. New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16(2), Apr. 1969.
- [11] G. Erkan and D. R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *J Artif Intell Res (JAIR)*, 2004.
- [12] M. Fiszman, T. C. Rindflesch, and H. Kilicoglu. Abstraction summarization for managing the biomedical research literature. pages 76–83, May 2004.
- [13] F. Giannetti. An exploratory mapping strategy for web-driven magazines. In *Proceeding of the eighth ACM symposium*, pages 223–229, New York, New York, USA, 2008. ACM Press.
- [14] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. pages 362–370, May 2009.
- [15] J. Hailpern, N. Damera-Venkata, and M. Danilevsky. Pagination: It's what you say, not how long it takes to say it. In *DocENG*. ACM, 2014.
- [16] J. Hailpern and B. A. Huberman. Echo: the editor's wisdom with the elegance of a magazine. In *EICS*. ACM Request Permissions, June 2013.
- [17] M. A. Hearst. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1), Mar. 1997.
- [18] N. Hurst, W. Li, and K. Marriott. Review of automatic document formatting. In *DocEng*, page 99, New York, New York, USA, Sept. 2009. ACM Request Permissions.
- [19] C. Jacobs, W. Li, E. Schrier, D. Barger, and D. Salesin. Adaptive grid-based document layout. *SIGGRAPH*, 22(3):838–847, July 2003.
- [20] N. Jamil, J. Mueller, C. Lutteroth, and G. Weber. Extending Linear Relaxation for User Interface Layout. In *ICTAI*. IEEE Computer Society, Nov. 2012.
- [21] I. Kastner and C. Monz. Automatic single-document key fact extraction from newswire articles. In *EACL*. Association for Computational Linguistics, Mar. 2009.
- [22] R. Katragadda, P. Pingali, and V. Varma. Sentence position revisited: a robust light-weight update summarization baseline algorithm. pages 46–52, June 2009.
- [23] C.-Y. Lin and E. Hovy. Identifying topics by position. In *ANCL*. Association for Computational Linguistics, Mar. 1997.
- [24] J. Liu, E. Wagner, and L. Birnbaum. Compare&contrast: using the web to discover comparable cases for news stories. In *WWW*, page 541, New York, New York, USA, May 2007. ACM.
- [25] C. Lutteroth, R. Strandh, and G. Weber. Domain Specific High-Level Constraints for User Interface Layout. *Constraints*, 13(3), Sept. 2008.
- [26] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *EMNLP*, 2004.
- [27] A. Nenkova. Automatic text summarization of newswire: lessons learned from the document understanding conference. In *AAAI*. AAAI Press, July 2005.
- [28] A. Nenkova and L. Vanderwende. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research, 2005.
- [29] L. Pevzner and M. A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 2002.
- [30] H. Schütze and C. Silverstein. Projections for efficient document clustering. In *SIGIR*, pages 74–81, New York, New York, USA, Dec. 1997. ACM Request Permissions.
- [31] A. Scoditti and W. Stuerzlinger. A new layout method for graphical user interfaces. In *TIC-STH*, pages 642–647. IEEE, 2009.
- [32] Y. Seki, K. Eguchi, and N. Kando. Compact Summarization for Mobile Phones. *Mobile and Ubiquitous Information Access*, 2954(Chapter 13):172–186, 2004.
- [33] J. Seo and W. B. Croft. Unsupervised estimation of dirichlet smoothing parameters. In *SIGIR '10*, pages 759–760, New York, New York, USA, 2010. ACM Press.
- [34] T. A. van Dijk. *News as discourse*. Lawrence Erlbaum Associates, Inc, 1988.
- [35] T. Weninger, W. H. Hsu, and J. Han. CETR: content extraction via tag ratios. *WWW 2010*, 2010.
- [36] C. C. Yang and F. L. Wang. Automatic summarization of financial news delivery on mobile devices. In *WWW'03*, 2003.
- [37] C. C. Yang and F. L. Wang. Hierarchical summarization of large documents. *J. of the American Society for Information Science and Technology*, 59(6), Apr. 2008.
- [38] C. Zeidler, J. Müller, C. Lutteroth, and G. Weber. Comparing the usability of grid-bag and constraint-based layouts. In *OzCHI*, pages 674–682, New York, New York, USA, Nov. 2012. ACM Request Permissions.
- [39] C. Zhai. *Statistical Language Models for Information Retrieval*. Morgan & Claypool Publishers, 2009.