Journal of Universal Computer Science, vol. 19, no. 3 (2013), 383-405
submitted: 7/2/12, accepted: 28/1/13, appeared: 1/2/13 © *J.UCS*

Text Representation for Efficient Document Annotation

Christin Seifert
(Passau University, Germany
christin.seifert@uni-passau.de)

Eva Ulbrich
(Know-Center Graz, Austria
eulbrich@know-center.at)

Roman Kern
(Know-Center Graz, Austria
Graz University of Technology, Austria
rkern@know-center.at)

Michael Granitzer
(Passau University, Germany
michael.granitzer@uni-passau.de)

Abstract: In text classification the amount and quality of training data is crucial for the performance of the classifier. The generation of training data is done by human labellers - a tedious and time-consuming work. To reduce the labelling time for single documents we propose to use condensed representations of text documents instead of the full-text document. These condensed representations are key sentences and key phrases and can be generated in a fully unsupervised way. We extended and evaluated the TextRank algorithm to automatically extract key sentences and key phrases. For representing key phrases we propose a layout similar to a tag cloud. In a user study with 37 participants we evaluated whether document labelling with these condensed representations can be done faster and equally accurate by the human labellers. Our evaluation shows that the users labelled tag clouds twice as fast and as accurately as full-text documents. While further investigations for different classification tasks are necessary, this insight could potentially reduce costs for the labelling process of text documents.

Key Words: document labelling, tag clouds, word clouds, text summarisation, data mining, supervised learning, TextRank
Category: H.1.7, H.1.2

1 Introduction

Text classification is a common task in data mining and knowledge discovery, applications include document organisation and hierarchical classification of web pages [Sebastiani, 2002]. Text classification is supervised learning, i.e., the classifier is built on the basis of a training data set. The training data consists of data items and one or more category labels for each of the data items. In general, the quality and amount of training data has great influence on the performance of the final classifier [Duda et al., 2000]. The generation of training data is usually done manually by domain experts. This means that data items are presented to domain experts, who manually assign class labels for each item - a repetitive, time consuming work. Approaches to reduce the overall labelling time can be grouped into approaches to reduce the amount of necessary training data and approaches to reduce the time for labelling a single training data item. The former include active learning [Settles, 2010] and semi-supervised learning strategies [Zhu, 2008]; an example of the latter is the "labelled feature approach" [Druck et al., 2008]. We follow the second route by reducing the time required to label single training items for text classification.

In this paper, we apply and extend automatic text summarisation methods and present an extensive user evaluation to test the suitability of the summaries for text classification. More specifically, we use key sentences and key phrases as compressed representations, both of which can be automatically extracted from text documents using the TextRank algorithm [Mihalcea and Tarau, 2004]. We extend the original TextRank algorithm and evaluate it on the data sets used in the original TextRank paper. We perform a user evaluation to investigate whether the developed representations (key sentences and key phrases) reduce the labelling time for a single document while guaranteeing that the category is still identifiable. We compare these representations to the baseline of text documents represented as full-text. From our user evaluation we conclude that labelling key phrases is not only twice as fast as labelling full-text documents but also as accurate. Further, the evaluation reveals that users do not believe that their decision is always correct, a fact that has to be considered when applying the method in applications. This issue is discussed in detail in section 5.2.4. The present work extends our previous work [Seifert et al., 2011] as follows:

1. We present a theoretical background to motivate the work.
2. We provide algorithmic details and a corresponding evaluation of our extensions to the TextRank algorithm.
3. We analyse the results from questionnaires and analyse their impact on the user studies. This shows that for practical applications it is necessary to resolve the discrepancy between users being most accurate using tag clouds and feeling that their decisions are incorrect.
4. We discuss the user suggestions for improving the manual document annotation process.

The remainder of this paper is structured as follows: Section 2 introduces the theoretical background and the research hypothesis. Section 3 discusses related work for minimising annotation time, text summarisation, and keyword layout. We extended the original TextRank algorithm to generate key words and key sentences. The original algorithm and our extension is described and evaluated in section 4. Section 5 presents the methodology for efficient user-based document annotation, including a user evaluation in Section 5.1 and its results in Section 5.2. Finally, Section 6 concludes the paper and indicates directions for future work.

2 Research Hypothesis

An average US-American reader is capable of reading 250-300 words per second (wps) on printed paper [Bailey, 1996]. Reading texts from computer displays is even slower [Ziefle, 1998]. The number of words has been shown to linearly influence the time for judging the relevance of documents for a search request [Jethani and Smucker, 2010]. More specifically, the time for the decision is $t = sw + k$, where w is number of words in the document, s the scan rate in seconds per word, and k is a constant overhead for making a decision. Although this model was developed for judging search results, we argue that it is applicable to text classification with a minor modification: The judgement of relevance is a binary decision, whereas in text classification the user has to decide between multiple classes, generally more than two. According to Hick's Law [Hick, 1952], the decision time increases logarithmically with the number of possible decisions. In classification, the number of possible decisions equals the number of classes. Thus, the constant overhead factor k needs to be changed to the logarithm of the number of classes. If we aim to reduce the overall time for the decision there are two choices according to the model: (i) increase the scan rate or (ii) decrease the number of words per document. The former requires training on the users' side, for instance by learning speed reading techniques [O'Brien, 1921, Abela, 2007], which is not feasible in most practical settings. The latter requires methods to automatically remove irrelevant words from texts, i.e. automatic text summarisation and keyword extraction. While the compression factor, the ratio of the original text length to the length of the summary, can be measured, the quality of the summary remains an open issue. Even with a comparison to annotated text summarisation corpora, we can not answer the question whether the summary is helpful for the task of efficient document annotation for text classification.

Given this theoretical background, our hypothesis is that commonly used tag cloud representations allow to comprehend text faster for taking a decision to which class a text belongs than standard full-text representations.

3 Related Work

In the field of *Machine Learning*, active learning is the most prominent approach to reduce the overall number of required training data [Settles, 2010]. In active learning the learning algorithm itself selects the most beneficial unlabelled item and updates its classification hypothesis using the label provided by the user [Settles, 2010]. Active learning aims at minimising the number of training samples to be labelled and thus reducing the overall labelling time. However, there is evidence, that (i) sequential active learning may increase the number of required training samples [Schein and Ungar, 2007], and (ii) batch-mode active learning may also require more training samples than random sampling. Furthermore, Tomanek & Olsen [Tomanek and Olsson, 2009] found out in their web survey that some experts in the natural language processing community do not trust active learning to work. The research of Baldridge & Palmer [Baldridge and Palmer, 2009] showed that the experience level of the annotator has an impact on the efficiency of the active learning approach. In their experiments, expert annotators performed best with uncertainty-based active learning, while non-expert annotators achieved better results using random sampling.

While active learning minimises the number of training documents, our goal is to minimise the time the user needs for identifying the category of a single document. Thus, active learning and our condensed text representations can be easily combined. Another work for minimising the time required for labelling single items was done in [Druck et al., 2008]. The authors showed that using labelled features, i.e. single words, instead of labelled text documents resulted in better classifier accuracy given limited labelling time. However, their approach is tailored towards a specific learning algorithm which may not be the algorithm of choice for a given text classification task. In contrast to their work, our approach is classifier agnostic, we efficiently generate a set of training documents that can then be used to train any classification algorithm.

Text Summarisation aims at producing a shorter version of the text while retaining the overall meaning and information content. In [Gupta and Lehal, 2010] a review for extractive summaries from texts is presented. Extractive summaries are a selection of meaningful document parts, while abstractive summaries are shorter rephrasings of the text. We use an enhanced version of the TextRank algorithm [Mihalcea and Tarau, 2004], as it allows for text summarisation on two different levels of granularity by extracting (i) key sentences and (ii) key phrases. The original TextRank algorithm has been improved by following the intuition of mutual reinforcement of words and sentences, which has already been applied on text summarisation [Zha, 2002].

Also the field of *Information Visualisation* offers ideas on alternative text representations [Šilić and Bašić, 2010]. Most of the visualisations show additional aspects of the text which are not instantly accessible in full-text representations. The Word Tree [Wattenberg and Viégas, 2008] for example, is a application of a keyword-in-context method and visualises word concordances. In TextArc [Paley, 2002] word frequencies and distributions of all words in the text are visualised. These visualisations allow to interactively investigate and explore the texts, but are neither condensing the text nor designed as topical summaries. PhraseNet [van Ham et al., 2009] shows inter-word relations and may be considered as a condensed visualisation of a text as two occurrences of the same phrase are collapsed into one node in the graph. True visual text summarisations are word clouds, such as Wordle [Viégas et al., 2009], or the Document Cards visualisation [Strobelt et al., 2009]. The latter on also resembles a normal word cloud in absence of tables or images in the documents. We use a special layout algorithm for displaying our word cloud [Seifert et al., 2008], which has the following properties: (i) the words are all displayed horizontally for better readability, (ii) the most important words are in the centre of the visualisation, (iii) there is no line-by-line alignment of the single words. We think that this special layout best resembles the nature of the extracted key phrases: There is a relation between the extracted key phrases because they originate from the same text, but the nature of the relation is unclear and the information of the sequence is lost.

4 TextSentenceRank Algorithm

This section describes and evaluates the TextSentenceRank algorithm – an extension of the TextRank algorithm [Mihalcea and Tarau, 2004]. The original paper of the TextRank algorithm proposes the usage of the algorithm to either detect key phrases or key sentences. For our implementation we combined both approaches into a unified method, the *TextSentenceRank*. In the following, we describe the TextRank algorithm (Section 4.1, our extension (Section 4.2) and an evaluation (Section 4.3)).

4.1 TextRank Algorithm

The TextRank algorithm is a graph-based ranking algorithm for determining a ranking of graph nodes. In short, the relevance of a node in the graph is determined by a voting mechanism. All predecessor nodes vote for a specific node, the score of a node is calculated from the scores of its predecessors. The final score for all nodes is determined by iteratively calculating the score for each node until the algorithm converges, i.e. the scores of all nodes are sufficiently stable. For keyword extraction, the nodes represent keywords and for key sentence extraction the nodes represent key sentences.

Consider a directed graph $G = (V, E)$ with V the set of nodes and E the set of edges. For a node v_i, $I(v_i)$ is the set of predecessor nodes, and $O(v_i)$ is the set of successor nodes. The score s of a node vi_i is then defined as

$$s(v_j) = (1 - d) + d \sum_{j \in I(v_i)} \frac{1}{|O(v_i)|} s(v_j) \tag{1}$$

with $d \in [0, 1]$ being a parameter accounting for latent connection between non-adjacent nodes. d models the probability of jumping from one node to another random node. The parameter d is set to 0.85 in the original TextRank implementation. Intuitively, the score of a node is defined by the score of the predecessor nodes, where a predecessor distributes its total score evenly to all its successors.

For the graph-based ranking for graphs constructed from texts, the connection between two links might be weighted. For weighted directed graphs, the score of a node v_i is given by $s(v_i)$ as follows:

$$s(v_i) = (1 - d) + d \sum_{j \in I(v_i)} \frac{w_{ij}}{\sum_{v_k \in O(v_j)} w_{jk}} s(v_j) \tag{2}$$

with w_{ij} being the weight of the edge between v_i and v_j.

Given the formulas for calculating the score of a single node, the graph-based ranking algorithm is given in algorithm 1. The iterative algorithm stops, if the total difference of subsequent scores is below a given threshold θ. The final scores of the nodes then define the ranking of the nodes.

Algorithm 1: TextRank algorithm for weighted graphs

randomly initialize $s(v_i)$ for all $v_i \in V$;
repeat
  foreach $v_i \in V$ **do**
    calculate $s'(v_i)$ using formula 2;
    calculate error $e_i = s'(v_i) - s(v_i)$;
    set $s(v_i) = s'(v_i)$;
  end
until $\sum_i e_i < \theta$;

This iterative approach of the original paper of the algorithm is only one way to calculate the final scores of the nodes. Another approach is based on the weighted adjacency matrix and the idea of eigenvector centrality. A description of this solution for the PageRank algorithm [Brin and Page, 1998] can be found in [Bryan and Leise, 2006]. In fact, TextRank is just an application of the PageRank algorithm for single text documents. In our implementation detailed in the next section, we use the latter approach.

For *extracting key sentences* the graph is constructed as follows: a node is created for each sentence. An edge between two nodes is created if their sentences are similar to each other, for instance in terms of cosine similarity of their feature vectors. On this weighted, undirected graph the graph-based ranking algorithm is applied. After the algorithm has converged graph nodes are sorted according to their score and the topmost nodes are selected.

For *extracting keywords* the graph is constructed as follows: (i) the text is tokenised, (ii) part-of-speech tags are assigned to each token, (iii) a node is created for all tokens with a specific part-of-speech tag, (iv) a link between two nodes is created if the words co-occur within a given window. On this unweighted, undirected graph, the graph-based ranking algorithm is applied. After the algorithm has converged, the nodes are sorted according to their score and the top T words are taken for post-processing. In the post-processing step, sequences of adjacent keywords are collapsed to multi-word keywords also termed key phrases.

4.2 Extension

The original paper of the TextRank algorithm proposes the usage of the algorithm to either detect key phrases or key sentences. For our implementation we combined both approaches into a unified method. The main intuition is similar to the mutual reinforcement principle formulated in [Zha, 2002], where key words boost key sentences and vice versa. To build the input graph for the eigenvector computations, we start with a graph built out of the words, identical to the original key phrase extraction approach. Next, we add nodes for all sentences and connect each sentence with all its words. The weight of the edges between sentences and words is set 50% higher than between adjacent words. For the first sentence we use a weight twice as much as the inter-term weights. This special weighting strategy has been implemented following the observation that the initial sentence often reflects the main topic of a document or paragraph [Hutchins, 1987].

By following this approach, both key phrases and key sentences are computed at the same time. The final eigenvector contains values for words as well as sentences. For the key phase detection, only those values are used that represent word nodes in the input graph. The same is done accordingly for sentences. The output of the modified TextRank algorithm is a sorted list of key words as well as key sentences. After applying a threshold to restrict the number of key words, adjacent key words can be combined to form key phrases like in the original proposal of the TextRank algorithm. No post-processing is necessary for the extraction of key sentences.

4.3 Evaluation

In order to assess the quality of the modified TextRank algorithm, a series of evaluations has been conducted independently from the rest of the system. The evaluation is based on the data-set provided by [Hulth, 2003], which also has been used in the evaluation of the original TextRank algorithm. We took the 500 test documents from this data-set to compute the precision and recall of the key phrase extraction. A number of manually picked key phrases are available for each document from the test set. On average for each document there are 9.8 key phrases, which occur at least once within the text.

The evaluation has been conducted in two stages. The first stage consists of a comparison of our implementation of the TextRank algorithm with the evaluation results presented in the original publication [Mihalcea and Tarau, 2004]. We wanted to make sure that we could successfully reproduce their results with our implementation of the algorithm. In Table 1, this comparison can be found in the first two rows. Although the numbers are not identical, the differences are not pronounced and it can be assumed that our implementation is sufficiently close to the original version. The performance also depends on factors outside the core key phrase extraction algorithm, like for example the preprocessing of the evaluation documents. When applying an older version of the library[1] which is used to split the plain text into words, we noticed a drop in the F_1 measure of about 0.3%.

In the second stage we evaluated the impact of the integration of the sentence information on the performance. The results are shown in Table 1 in the rows for *TextRank* and *TextSentenceRank*. The enhanced version of the algorithm did improve the F_1 measure by about one percent. This difference can only be regarded as minor improvement.

The default configuration of the original TextRank algorithm produces a relatively high number of key phrases for each document. In our use-case we want to achieve a low number of key phrases, which still cover the main topical aspects. Therefore we restricted the number of key words to $\sqrt{n}$, where n is the number of unique nouns and adjectives in the document. This results in the extraction of about just a third of number of key phrases in comparison with the original threshold. In the generation of the word clouds we always applied the lower threshold resulting in fewer key phrases.

In the last two rows in Table 1 the performance of our key phrases detection algorithm is represented when using the lower threshold. The performance does drop dramatically when generating fewer key phrases. In this setting the *TextSentenceRank* algorithm produces far better results than the original algorithm. This indicates that the sentence information is far more valuable as has been apparent from the first evaluation runs.

The main reason for the drop in performance in the setting with a lower threshold is the way how key phrases are generated. Due to fewer single key words being selected, the chances for adjacent words to be combined into key phrases is reduced greatly. It is only assessed in the evaluation whether a complete key phrase has been extracted. Partial matches are counted as misses. Whether the relatively low performance in this evaluation does truly reflect the usefulness of the approach is addressed in the user evaluation.

[1] OpenNLP 1.5.1 vs. 1.4.3

	Key Phrases		$\boldsymbol{\pi}$	$\boldsymbol{\rho}$	$\mathbf{F_1}$
	Extracted	**Average**			
TextRank$^{\text{orig}}$	6,784	13.7	31.2%	43.1%	36.2%
TextRank	6,379	12.8	31.9%	41.4%	36.0%
TextSentenceRank	6,782	13.6	32.0%	44.1%	**37.1**%
TextRank$^{\text{low}}$	2,232	4.5	6.4%	2.9%	4.0%
TextSentenceRank$^{\text{low}}$	1,993	4.0	18.4%	7.5%	**10.6**%

Table 1: Key word extraction performance. Comparing original TextRank algorithm (*TextRank*$^{\text{orig}}$), our implementation (*TextRank*) and the version, which incorporates the sentence information into the process of key phrase detection (*TextSentenceRank*). Comparing precision $\boldsymbol{\pi}$, recall $\boldsymbol{\rho}$ and $\mathbf{F}_1$ measure. The last two rows show the performance of the key phrase extraction, if only a small fraction of all words are selected as key words.

5 Approach for Efficient Document Annotation

This section presents the methodology to evaluate the effect of different text representations on manual labelling speed and accuracy. Figure 1 gives an overview of the methodology. Starting from text documents (on the left) three different paths for generating the three different text representation forms are shown. In this paper we use the word "condition" as a synonym for the text representation form, because each text representation form resembles a condition in our user evaluation. The three different conditions are denoted as **F** for full-text, **S** for key sentences (and named entities) and **P** for key phrases.

For *pre-processing* we used a standard information extraction pipeline based on the OpenNLP[2] library. The pipeline consists of the following steps: tokenisation, stemming, stop-word removal, part-of-speech tagging and named entity extraction. We used a set of standard German stop words and the German language model for part-of-speech tagging. In general the proposed is applicable to any language given the language dependent models for preprocessing. The keywords and key sentences were extracted using the TextRank extension described in section 4. The named entities of type "person" were added to the extracted key phrases and together they represent the key phrase condition **P** in the experiments.

Concerning the user interface, the *layout* for the full-text and the key sentences is straightforward by using a simple text window. The key phrases extracted by the TextRank algorithm may originate from any location of the source

[2] `http://incubator.apache.org/opennlp/`

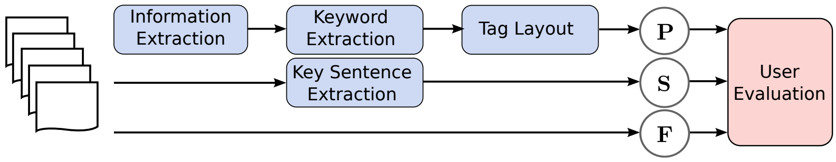

Figure 1: Overview of the methodology: three different document representations are generated and compared in a user evaluation.

text. Two key phrases may belong to the same sentence and share the same context but they also may not. Consequently two key phrases have a relation as they are extracted from the same text but we do not know which type of relation it is. We chose to use a layout for the key phrases and named entities that reflects this uncertainty in the relations. A line-by-line (Western reading-direction) layout would indicate either a relation in reading direction between the words, or none relation at all for people used to read tag clouds. We chose a layout algorithm from the family of tag layout algorithms described in [Seifert et al., 2008], where the words are laid out in a circular manner, starting from the centre-of-mass of the visualisation boundary. The interesting property of this layout algorithm for our use case is that words are not aligned on a line and thus reading line-by-line is not possible. Compared to other word clouds, such as those generated by Wordle [Viégas et al., 2009] the words are still easily readable, because all words are aligned horizontally.

5.1 User Evaluation

In the user evaluation we wanted to examine whether the text representation form (full-text, key sentences, key phrases) had an influence on the correctness of the labels assigned to the documents and the time required for labelling. Moreover we wanted to examine the influence of the potential wrong labels on different classifiers. In particular we tested the following hypotheses:

H1 The time required for labelling key phrases or key sentences is significantly less than for labelling full-text documents.

H2 There is no difference in the number of correct labels between key phrases, key sentences and full-text.

H3 There is no difference in classifier accuracy when using labels generated in the key phrases, key sentences or full-text condition.

Furthermore, we were interested in the users' perception of their performance. Also we wanted to find out whether they preferred a particular representation form or disliked another one. More specifically, beyond the hypotheses enumerated above we investigated the following assumptions:

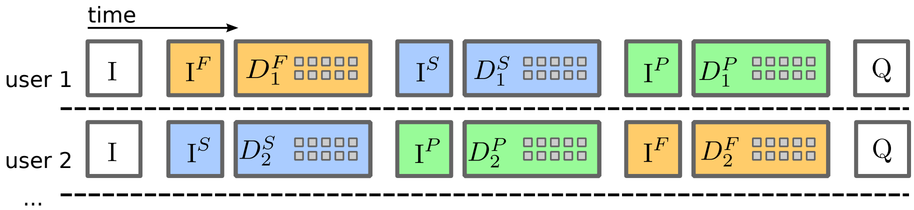

Figure 2: Overview of the evaluation procedure, I abbreviates an Introduction step, **F** (full-text), **S** (key sentences), and **P** (key phrases) denote the different conditions. Q abbreviates the final questionnaire.

A1 Users prefer the key sentence representation to the other two, because its more familiar than the word clouds and more concise than the full-text representation.

A2 Users feel deciding most accurately using the full-text representation, because it contains most information.

5.1.1 Design

We used a within-subjects design. The independent variable is the text representation form with three different levels (full-text **F**, key sentences **S** and key phrases **P**). We measured task completion time and correctness (dependent variables). The task completion time is measured as the time difference between the user first seeing the document and finishing the assignment for this document. Correctness of the task is calculated as the number of correct user labels by comparing the user labels to the ground truth of the annotated corpus.

5.1.2 Procedure

Figure 2 gives an overview of the evaluation procedure. For each participant, the study started with an introduction of the task and with an example document for each condition. Then the participant had time to ask questions. Thereafter the participant was asked to fill out a demographic questionnaire. Then, the three trials on the computer started. The sequence of conditions (**F**, **S** and **P**) and the documents were randomly chosen from the data set (see section 5.1.3 for details). For one trial (10 subsequent documents) the presentation form was the same (e.g., all documents presented as full-text). Each trial started with an introductory screen. After the participant had clicked "OK", the measurements started. We measured the task completion time (the time between the two subsequent clicks on the "OK" button) and collected the labels that the participants assigned to the articles. For each of the three conditions, we computed the mean value for the completion time and counted the number of correct labels. Thus, for each participant $i, 1 \leq i \leq 37$ we obtained one single value for the number of correct labels l_i^c, and completion time t_i^c per condition $c \in \{\mathbf{F}, \mathbf{S}, \mathbf{P}\}$. The study finished with a questionnaire for the participants were we asked questions about the perceived overall task difficulty, and the stress. Further, we asked the participants to rate the perceived helpfulness, speed and difficulty and how much they liked to work with the specific representation. The users rated these aspects on a 5 point Likert scale. In the following the questions are listed (translated from German) and the meaning of the different values is noted in brackets (X is place holder for the different representation forms).

a) How helpful was X for finding the correct topic? (1 - not, .., 5 - very)
b) How difficult was it to find the correct topic with X? (1- very, .., 5 - not)
c) How fast could you identify the correct topic with X? (1 - slow, .., 5 - fast)
d) Was it stressful to identify the topics using X? (1 - very, .., 5 - not)
e) How much did you like to use X? (1 - not, .., 5 - very much).
f) Were there enough hints for the topics in X? (1- too few, ..., 5 - enough)
g) How difficult was it, to map the texts to the topics? (1-very, .., 5 - not)
h) How stressful was the task in general? (1 - very, .., 5 - not)

Further we asked the participants two open questions: (i) Which text representation forms could you imagine to make texts accessible in a fast way? (ii) How could we improve the presented text representation forms in your opinion? Here the users were not given any choices to select from but could write their answer in free-form text.

5.1.3 Test Material

We used a German news corpus from the Austrian Press Agency consisting of 27570 news articles from the year 2008. We chose news articles from 2008 to reduce the effect that users simply remembered recent news. The corpus is fully labelled, i.e., each news article is annotated with one of the five classes "economy, sports, culture, politics, science“. The articles are nearly equally distributed over the classes. The length of the articles varies between 2 and 2720 words, the average length is 247.2 words. We investigated how often the class names occur in the extracted key words. For 616 of 27570 ($\approx 2\%$) of the documents an extracted key phrase is equal to a word describing the class. This means, it is very unlikely that users can use the class name as a clue for labelling.

We chose the longest articles of the corpus, i.e. the articles longer than the 3rd quantile (> 337 words) without the statistical outliers (articles with > 655 words). This leaves 6328 articles for the experiment, 1508 in class "culture“, 1023 in "economy“, 1409 in "politics“, 1457 in “science” and 931 in "sports“.

For each condition a set of 10 documents is presented to the user. The document set for a condition is denoted as $D^{\mathbf{F}}$, $D^{\mathbf{S}}$, $D^{\mathbf{P}}$ respectively. For a user k the sets are denoted as $D_k^{\mathbf{F}}$, $D_k^{\mathbf{S}}$, $D_k^{\mathbf{P}}$. All articles in all document sets are distinct, i.e., no user gets a document twice. For articles in set $D^{\mathbf{S}}$ key sentences, for articles in set $D^{\mathbf{P}}$ key phrases and named entities were extracted as described in section 4. The key sentences and the full-text were displayed in standard text windows. The key phrases and named entities were laid out as tag cloud. In order to visually separate key phrases and named entities, the key phrases were coloured black and the named entities were coloured blue. An example for a key phrases representation is shown in Figure 3.

Figure 3: Example word cloud for the key phrases condition **P**.

5.1.4 Participants and Environment

37 German-speaking volunteers participated in the evaluation, 18 females and 19 males. 23 of the participants were technical professionals while 14 were experts of other domains. The age of the participants ranged from 25 to 58 years (average 32.5 years). The participants were tested in a calm environment without noise distractions or additional attendees. The task was performed on a Dell Latitude e650 notebook running Windows XP Professional. The notebook was equipped with an Intel Core Duo 2.26 GHz and 3 Gb RAM. The display resolution was 1440 x 900 pixels. All users were required to use the USB mouse (and not the touch pad).

5.2 Results and Discussion

In this section we present (i) how we tested the three hypotheses enumerated at the beginning of Section 5.1 (measured performance), (ii) the results of the quantitative evaluation of the questionnaire (perceived performance), (iii) answers to the questionnaire's open part (suggestions and improvements).

	full-text	key sentences	key phrases
labels	7.84 ± 1.24	7.59 ± 1.38	8.24 ± 1.23
time [s]	19.9 ± 13.8	10.7 ± 4.4	10.4 ± 4.1

Table 2: Overview of task completion time and number of correct labels (out of 10) for each condition. Values averaged over all users, showing mean and standard deviation.

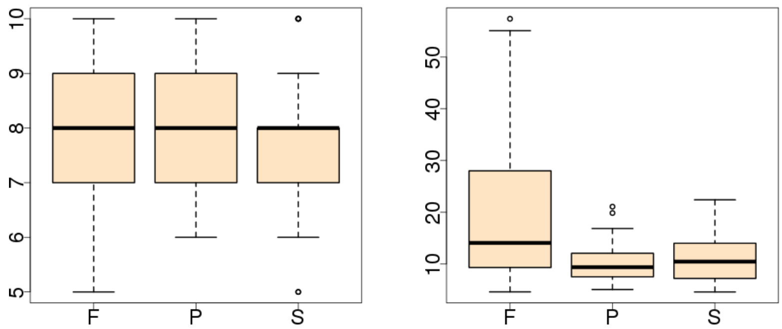

Figure 4: Box plots for (left) number of labels (correct out of ten) and (right) task completion time. Averaged over all users.

5.2.1 Measured Performance

Table 2 and Figure 4 summarise the measures for the number of correctly labelled examples and the task completion time. Altogether, the users assigned 290 correct labels in the full-text condition, 281 in the key sentences condition and 305 in the key phrases condition. In total 370 documents (10 documents per user, 37 users) were labelled in each condition. In the following sections we describe in detail how we tested the three hypotheses enumerated at the beginning of Section 5.1.

Influence on Labelling Accuracy: We tested whether the difference in the correct number of labels reported in Table 2 are significant (Hypothesis H2). The correct number of labels is denoted as l_i^c for person i and condition c. We tested the variables l^f, l^s and l^p for normal distribution using the Shapiro-Wilks test [Shapiro and Wilk, 1965]. All variables are not normally distributed, assuming $\alpha < .05$, thus the precondition for performing ANOVA or paired T-tests is not satisfied. Therefore, we tested on equal means with Wilcoxon rank sum test for unpaired samples. The null hypothesis for the test was that the means are equal, we set $\alpha = .05$. No difference in the mean values was found between full-text and key phrases ($W = 563$, $p = .177$) and between full-text and key sentences ($W = 754$, $p = .441$). Comparing key sentences to key phrases we found a significant difference in the mean values ($W = 504$, $p = .46$).

Summing up, we found out that users assigned significantly less correct labels when using the key sentence representation of the documents, but performed equally well with the full-text representation and the word cloud.

Influence on Labelling Time: We tested further whether the differences in task completion time reported in Table 2 are significant (Hypothesis H1). The average time for labelling is denoted as t_i^c for person i and condition c. We tested the variables t^f, t^s and t^p for normal distribution using the Shapiro-Wilks test. All variables are not normally distributed, assuming $\alpha < .05$, thus the condition for performing ANOVA or paired T-tests is not satisfied. Therefore, we tested on equal means with Wilcoxon rank sum test for unpaired samples. The null hypothesis for the test was that the means are equal, we set $\alpha = .05$. No difference in the mean values was found between the full-text and key sentences ($W = 705$, $p = .830$). On the contrary, we found a significant difference comparing full-text and key phrases ($W = 956$, $p = .003$) and full-text and key sentences ($W = 982$, $p = .001$).

Summing up, we found out that users labelled the items significantly faster when using the key sentence or the key phrases representation than when using the full-text representation of the documents.

Influence on Classifier Accuracy: As reported at the beginning of this section we found out that users labelled less accurately when using the key sentence representation of the text documents. We further wanted to test, whether this mislabelling would have an influence on classifiers trained on the erroneous labels (Hypothesis H3). To do so, we created two different training data sets for each condition, resulting in 6 different training data sets. Both training sets for one condition contained the documents processes by all users in this condition, one was extended by the original labels (the ground truth) and the other one was extended by the user labels. We further created an evaluation data set of 6000 randomly selected items from the data set. None of the evaluation items was contained in any of the training data sets. We trained various classifiers on both training data sets for each condition, and evaluated the trained classifiers on the evaluation data set. a_o^c denotes the accuracy of the classifier trained on original labels, a_u^c denotes the accuracy of the classifier trained on user labels for condition c. We applied the following classifiers:

(a) AdaBoost with Decision Stumps, implementation from the Mallet machine learning library [McCallum, 2002] with default parameters,
(b) Bagging with Decision Stumps from the Mallet machine learning library using default parameters,
(c) Naive Bayes from the WEKA machine learning library [Hall et al., 2009],
(d) Hyperpipes from the WEKA machine learning library,

classifier	full-text a_o^f	a_u^f	key sentences a_o^s	a_u^s	key phrases a_o^p	a_u^p
KNN-10	0.76	0.72	0.77	0.73	0.76	0.73
Bagging-DT	0.45	0.45	0.51	0.48	0.47	0.45
LibLin	0.80	0.74	0.80	0.76	0.79	0.74
KNN-20	0.75	0.71	0.76	0.73	0.76	0.72
Adaboost-DT	*0.36*	*0.41*	0.39	0.38	0.33	0.31
NaiveBayes	0.81	0.77	0.78	0.76	0.79	0.76
CFC, b=2.3	0.78	0.73	0.78	0.73	0.78	0.72
Hyperpipes	0.78	0.72	0.77	0.71	0.77	0.67

Table 3: Classifier accuracy when trained on original labels (a_o) versus trained on user labels (a_u) for the three different conditions full-text, key sentences and key phrases

(e) Linear Support Vector Machine from the LibLinear library [Fan et al., 2008] using default parameters,
(f) K-Nearest Neighbor classifier (own implementation), denoted KNN-10 for k=10, and KNN-20 for k = 20,
(g) Class-feature-centroid classifier [Guan et al., 2009] (own implementation), denoted CFC.

Table 3 reports the accuracy of the classifiers on the evaluation data set. Not surprisingly, the accuracy of the classifier trained on user labels was lower in nearly every case than when trained on the original (ground truth) labels. This is because the ground truth was labelled by domain experts and we did not explicitly communicate the rules for assigning an article to a specific category. Thus, for the boundary articles, i.e., news about a politician attending a sports event, the decision whether the article belongs to category "sports" or "politics" was subjective. Because all articles were randomly selected and aligned to the three conditions this effect is likely to occur equally often in all conditions. The one exception is the Adaboost classifier in the full-text condition. However, this is also the classifier that performs worst for this classification task.

Table 4 reports the differences in classifier accuracy averaged over all classifiers for the three conditions. When using the user-labels the accuracy decreases by less than 4% in all conditions. The difference in accuracy for the key phrases seems to be larger ($\Delta a^p = 0.040$) than for the sentence and full-text conditions ($\Delta a^s = 0.034$, $\Delta a^f = 0.034$). We investigated whether these differences are statistically significant. First we tested the variables Δa^f, Δa^s and Δa^p for normal distribution using the Shapiro-Wilks test ($\alpha = 0.05$). The two variables Δa^s and

	full-text	**key sentences**	**key phrases**
Δcorrect labels	71	80	65
Δa	0.034 ± 0.037	0.034 ± 0.017	0.040 ± 0.022

Table 4: Comparing original labels and user labels: Difference in number of correct labels and classifier accuracy (mean and standard deviation)

Δa^p follow a normal distribution, but Δa^f does not. This means, the preconditions for calculating ANOVA or paired t-Tests was not fulfilled. Therefore we used the Wilcoxon rank sum test for unpaired samples to compare the mean values using $\alpha = .05$. We found no significant difference between any of the conditions, the test statistics are as follows: full-text vs. key phrases $W = 39$, $p = .462$, full-text vs. key sentences $W = 34$, $p = .833$, key sentences vs. key phrases $W = 29$, $p = .753$.

To sum up, we found no influence of the different representation forms on classifier accuracy.

5.2.2 Perceived Performance

In this section the results of the qualitative and quantitative analysis of the questionnaire are presented.

Table 5 gives an overview of averaged values of the perceived helpfulness, speed, difficulty, stress, hints, and how much the user liked the respective text presentation form. A detailed description of the questions and the scale can be found in Section 5.1.2. Note, that the second column shows the values for the overall task, which the participants had to rate first, i.e. after the experiments there were firstly asked how difficult and stressful they perceived the experiment as a whole. From Table 5 we can draw the following conclusions:

(a) The *full-text* condition was perceived as most helpful and least difficult. Further, participants felt that it presented nearly all information they needed to complete the task (hints). Also full-text was liked the most.

(b) Participants perceived themselves as performing fastest in the *key phrases* condition, however, they were more stressed and liked it less than the other two conditions. Further, the *key phrases* condition gave them the least hints on the topic of the document.

(c) The *key sentence* representation caused least stress for the participants. Further, participants felt that they were nearly as fast in this condition as in the key phrases (fastest) condition.

question	overall task	full-text	key sentences	key phrases
helpfulness	–	**4.1 ± 1.0**	3.9 ± 1.0	3.4 ± 1.1
speed	–	3.6± 1.0*	3.9 ± 1.0	**4.0 ± 1.0**
difficulty	3.7 ± 0.6	**4.1 ± 1.0**	3.8 ± 1.2	3.4 ± 1.2
stress	4.1 ± 0.8	3.4 ± 1.3	**3.9 ± 1.0**	3.4 ± 1.3
hints		**4.7 ± 0.8**	3.8 ± 1.1	3.2 ± 1.1
like	–	**3.7 ± 1.2**	3.5 ± 1.4	3.4 ± 1.4

Table 5: Overview of questionnaire answers. Showing mean and standard deviation averaged over all participants. Best values (most hints, fastest, ..) for each row are marked bold. * one missing value in the data set replaced by the median

To sum up, we can conclude that users preferred full-text, were least stressed by the key sentences and found that they performed fastest in the key phrases condition.

5.2.3 Suggested Improvements

We asked open questions for suggestions on improving the representation forms or find substitutes as well as possible amendments. The question "How could the representation be improved?" was answered by 57% while 65% of the participants had suggestions on "Which other representation could be used to allow a quick understanding of information in texts?" As most of the answers headed into similar directions we categorised them. An overview of the categories, and the number of answers is shown in Table 6.

	Category	Description	#answers
Q1	*Format*	formatting issues, e.g. paragraphs, font, font sizes	14
	Content	represented content, e.g. words or sentences	3
	Structure	structure of the representation form, e.g. ordering	4
Q2	*Format*	formatting issues	12
	Representation	alternative representations and visual aids, e.g. tables or pictures	12

Table 6: Categories and number of answers for questions 1 (Q1) and 2 (Q2).

Examples answers for the first question concerning *Content* are : "formatting of the full text examples, paragraphs, font", "bigger font, different font". Category *Structure* of the first question contains suggestions like: "Key sentences: they are hard to read. Relations between the sentences are not logical". Category *Format* of the same question had answers including: "Group words by the kind of words at key phrases (named entities, names, adjectives)".

Example answers for category *Format* are: "Highlight keywords in long texts", "Formatting (breaks, fat, italic, colors!)". *Representation* contains suggestions like: "A combination between full-text and key phrases (more key phrases)", "Hypertree, TreeMap", and "Pictures, Graphics, Icons".

5.2.4 Discussion

In this section we discuss our hypotheses outlined at the beginning of Section 5.1 in the light of the results of the previous section. The evaluation showed that users can label key words twice as fast but with the same accuracy as full-text documents. Labelling of key sentences is fast too, but the labelling accuracy is significantly lower than in the full-text condition. This means we can accept hypotheses **H1**: that a compressed representation leads to faster decisions, regardless whether this decision is correct or not. Hypothesis **H2** must be rejected, there is a difference in the number of correct labels when varying the representation form. More specifically, users are most accurate when using full-text or key phrases, indicating that the TextSentenceRank algorithm for keyword extraction performs well in filtering out information irrelevant for text categorization while keeping the information required to identify the category. On the contrary, the labelling accuracy for key sentences is significantly lower, indicating that key sentences are less informative on average, obviously either irrelevant or ambiguous sentences are extracted. In our experiments we found no influence of this different labelling accuracy on classifier performance confirming hypothesis **H3**. This might be due to the noise tolerance of the used classifiers and the practically low amount of noise. In our experiment, it makes no difference for the classifier whether 65 or 80 out of 370 documents are labelled incorrectly. We expect this difference to become significant when the number of training items (and thus the number of mislabelled items) increases.

Analysing the questionnaire lead to interesting results. We found that on average users preferred the full-text to the key sentences and key phrases. Still they found key phrases more stressful than the key sentences and key phrases. We suppose that this is because, (i) they are used to read sequences of sentences (as opposed to word clouds) and (ii) the texts are shorter than full-text, i.e. less text to read in total. Further, users felt labelling fastest in the key phrases condition and nearly and slowest in the full-text condition. Thus, our assumption **A1** could only be partly confirmed.

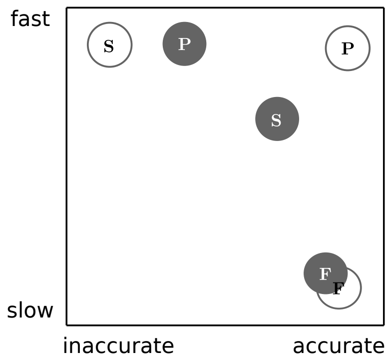

Figure 5: Infographics showing perceived (filled circles) and measured performance (empty circles) for key phrases (P), key sentences (S), and full-text (F).

Interestingly, in some aspects the perceived performance does NOT conform to the measured performance. First, users felt that they were least hints in the key phrases condition which was also perceived least helpful, but our measurements show, that they performed as accurately as in the full-text condition. Thus, our assumption **A2** can be confirmed. Second, users felt to have more hints in the key sentences condition and found the presentation more helpful than the key phrases condition, but according to the measurements performed worst in the key sentence condition. Note, that participants on average did like the key phrases condition less than the others. We assume that this is because the key phrases representation was unfamiliar. For applications this would mean that users need to be convinced of the helpfulness of the key phrases representation and get used to it. The info-graphics in Figure 5 summarises the differences between perceived and measured values. (Note that this info-graphics visualises tendencies, not accurate values).

The answers to the open questions gave interesting hints on how to improve representation of text as well as find new ideas on how to structure information in texts. The utmost answers to question one, how to improve the given representations, regarded formatting and highlighting. We therefore believe that a way to change font and size of texts like buttons to control the font size could be of much help to faster understand the text information. Furthermore, colouring keywords could help users to find relevant information faster. To satisfy the suggestions related to content, other ways to select words and phrases could be tested. As for the structuring of the representation forms most of the answers suggested a different ordering of key phrases. The ordering of the words could be made selectable, for example, nouns and names before other words. The second open question of the questionnaire showed that many users had similar ideas about highlighting important words in full-text. This could be met by giving users the possibility to automatically highlight all nouns or names. Users also had suggestions on how to combine different representation forms. Combinations of key phrases and full-text representations, i.e. showing full-text with highlighted key phrases could be useful. Also, interesting ideas emerged regarding visualizations and graphics. Tables containing words of a category or a graphical representation of the of the keywords count could be shown.

Summing up, our evaluation shows that: *Key phrases are a fast and accurate representation for document labelling. In fact, users labelled key phrases twice as fast and as accurately as full-text documents. Further, the questionnaire shows, that users did not trust their labellings with the word cloud representation, although they performed most accurately.*

6 Conclusion and Future Work

We investigated two different condensed representations of text, key phrases and key sentences, for the purpose of faster document labelling. Both representation forms can be generated in a fully automatic way. We propose and evaluate an extension to the TextRank algorithm. The idea of this extension is that key words occur more often in key sentences and as such words and sentences should boost each other's rank.

In a user evaluation we compared the labelling accuracy and time of the users when using these condensed representations to the baseline, the full-text representation of the texts. Our evaluation shows that the users labelled key phrases twice as fast but as accurately as full-text documents. This finding points toward a feasible way to decrease the time and cost for the generation of training data. Word clouds for labelling can be easily combined with other approaches such as active learning.

Further experiments are necessary to investigate the benefit for other classification tasks. Directions of experiments include: texts in other languages and hierarchical and/or multi-label classification problems. Further, the process of extracting the condensed information (keyword extraction) as well as the presentation (number of keywords to show, layout algorithm) can be varied. During the user evaluation we got the impression, that different users used different reading patterns ranging from sequential word-by-word reading to scanning. We plan an eye-tracking study to investigate to which extend the reading patterns influence the efficiency of the word cloud representation. Following this direction, an application can then implement a combined or adaptive user interface: the initial representation is the word cloud, once the user feels that the presented information is insufficient to identify the label she can request the full-text article.

Acknowledgements

This work has been funded by the European Commission as part of the TEAM IAPP project (grant no. 251514) within the FP7 People Programme (Marie Curie). The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[Abela, 2007] Abela, J. (2007). *X-treme Speed Reading.* Marshall Cavendish.

[Bailey, 1996] Bailey, R. (1996). *Human Performance Engineering: Designing High Quality Professional User Interfaces for Computer Products, Applications and Systems.* Prentice-Hall.

[Baldridge and Palmer, 2009] Baldridge, J. and Palmer, A. (2009). How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation. In *Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 296–305, Morristown, NJ, USA. Association for Computational Linguistics.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117.

[Bryan and Leise, 2006] Bryan, K. and Leise, T. (2006). The $25,000,000,000 eigenvector: The linear algebra behind google. *SIAM Rev.*, 48:569–581.

[Druck et al., 2008] Druck, G., Mann, G., and McCallum, A. (2008). Learning from labeled features using generalized expectation criteria. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602, New York, NY, USA. ACM.

[Duda et al., 2000] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification.* Wiley-Interscience, 2nd edition.

[Fan et al., 2008] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.

[Guan et al., 2009] Guan, H., Zhou, J., and Guo, M. (2009). A class-feature-centroid classifier for text categorization. In *Proc. of the International conference on World Wide Web (WWW)*, pages 201–210, New York, NY, USA. ACM.

[Gupta and Lehal, 2010] Gupta, V. and Lehal, G. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3).

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18.

[Hick, 1952] Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4:11–26.

[Hulth, 2003] Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 10(2000):216–223.

[Hutchins, 1987] Hutchins, J. (1987). Summarization: Some problems and methods. In Sparck Jones, K., editor, *Meaning: The Frontier of Informatics*, pages 26–27. ASLIB, London.

[Jethani and Smucker, 2010] Jethani, C. P. and Smucker, M. D. (2010). Modeling the time to judge document relevance. In *SIGIR Workshop on the Simulation of Interaction*, Genova.

[McCallum, 2002] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[Mihalcea and Tarau, 2004] Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

[O'Brien, 1921] O'Brien, J. A. (1921). *Silent reading, with special reference to methods for developing speed; a study in the psychology and pedagogy of reading.* The MacMillan company.

[Paley, 2002] Paley, W. (2002). Textarc: Showing word frequency and distribution in text. In *Proceedings of IEEE Symposium on Information Visualization, Poster Compendium*, IEEE CS Press.

[Schein and Ungar, 2007] Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Mach. Learn.*, 68(3):235–265.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

[Seifert et al., 2008] Seifert, C., Kump, B., Kienreich, W., Granitzer, G., and Granitzer, M. (2008). On the beauty and usability of tag clouds. In *Proceedings of the 12th International Conference on Information Visualisation (IV)*, pages 17–25, Los Alamitos, CA, USA. IEEE Computer Society.

[Seifert et al., 2011] Seifert, C., Ulbrich, E., and Granitzer, M. (2011). Word clouds for efficient document labeling. In *The Fourteenth International Conference on Discovery Science.*

[Settles, 2010] Settles, B. (2010). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

[Shapiro and Wilk, 1965] Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611.

[Šilić and Bašić, 2010] Šilić, A. and Bašić, B. (2010). Visualization of text streams: A survey. In Setchi, R., Jordanov, I., Howlett, R., and Jain, L., editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6277 of *Lecture Notes in Computer Science*, pages 31–43. Springer Berlin / Heidelberg.

[Strobelt et al., 2009] Strobelt, H., Oelke, D., Rohrdantz, C., Stoffel, A., Keim, D. A., and Deussen, O. (2009). Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics*, 15:1145–1152.

[Tomanek and Olsson, 2009] Tomanek, K. and Olsson, F. (2009). A web survey on the use of active learning to support annotation of text data. In *Proc. of the NAACL Workshop on Active Learning for Natural Language Processing (HLT)*, pages 45–48, Morristown, NJ, USA. Association for Computational Linguistics.

[van Ham et al., 2009] van Ham, F., Wattenberg, M., and Viegas, F. B. (2009). Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15:1169–1176.

[Viégas et al., 2009] Viégas, F. B., Wattenberg, M., and Feinberg, J. (2009). Participatory visualization with wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15:1137–1144.

[Wattenberg and Viégas, 2008] Wattenberg, M. and Viégas, F. B. (2008). The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14:1221–1228.

[Zha, 2002] Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*, page 113.

[Zhu, 2008] Zhu, X. (2008). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin.

[Ziefle, 1998] Ziefle, M. (1998). Effects of display resolution on visual performance. *Human Factors*, 40(4):555–568.