On the Helmholtz Principle for Data Mining

Alexander Balinsky, Helen Balinsky, Steven Simske

HP Laboratories
HPL-2010-133

Keyword(s):

extraction, feature extraction, unusual behavior detection, Helmholtz principle, mining textual and unstructured datasets

Abstract:

We present novel algorithms for feature extraction and change detection in unstructured data, primarily in textual and sequential data. Keyword and feature extraction is a fundamental problem in text data mining and document processing. A majority of document processing applications directly depend on the quality and speed of keyword extraction algorithms. In this article, a novel approach to rapid change detection in data streams and documents is developed. It is based on ideas from image processing and especially on the Helmholtz Principle from the Gestalt Theory of human perception. Applied to the problem of keywords extraction, it delivers fast and effective tools to identify meaningful keywords using parameter-free methods. We also define a level of meaningfulness of the keywords which can be used to modify the set of keywords depending on application needs.

External Posting Date: October 6, 2010 [Fulltext]   Approved for External Publication
Internal Posting Date: October 6, 2010 [Fulltext]

On the Helmholtz Principle for Data Mining

Alexander Balinsky
Cardiff School of Mathematics
Cardiff University
Cardiff CF24 4AG, UK
Email: BalinskyA@cardiff.ac.uk

Helen Balinsky
Hewlett-Packard Laboratories
Long Down Avenue
Bristol BS34 8QZ, UK
Email: Helen.Balinsky@hp.com

Steven Simske
Hewlett-Packard Laboratories
3404 E. Harmony Rd. MS 36
Fort Collins, CO USA 80528
Email: Steven.Simske@hp.com

***Abstract*—We present novel algorithms for feature extraction and change detection in unstructured data, primarily in textual and sequential data. Keyword and feature extraction is a fundamental problem in text data mining and document processing. A majority of document processing applications directly depend on the quality and speed of keyword extraction algorithms. In this article, a novel approach to rapid change detection in data streams and documents is developed. It is based on ideas from image processing and especially on the Helmholtz Principle from the Gestalt Theory of human perception. Applied to the problem of keywords extraction, it delivers fast and effective tools to identify meaningful keywords using parameter-free methods. We also define a level of meaningfulness of the keywords which can be used to modify the set of keywords depending on application needs.**

***Keywords*-keyword extraction, feature extraction, unusual behavior detection, Helmholtz principle, mining textual and unstructured datasets**

I. Introduction

Automatic keyword and feature extraction is a fundamental problem in text data mining, where a majority of document processing applications directly depend on the quality and speed of keyword extraction algorithms. The applications ranging from automatic document classification to information visualization, from automatic filtering to security policy enforcement – all rely on automatically extracted keywords [1]. Keywords are used as basic documents representations and features to perform higher level of analysis. By analogy with low-level image processing, we can consider keywords extraction as low-level document processing.

The increasing number of people contributing to the Internet and enterprise intranets, either deliberately or incidentally, has created a huge set of documents that still do not have keywords assigned. Unfortunately, manual assignment of high quality keywords is expensive and time-consuming. This is why many algorithms for automatic keywords extraction have been recently proposed . Since there is no precise scientific definition of the meaning of a document, different algorithms produce different outputs.

The main purpose of this article is to develop novel data mining algorithms based on the Gestalt theory in Computer Vision and human perception. More precisely, we are going to develop Helmholtz principle for mining textual, unstructured or sequential data.

Let us first briefly explain the Helmholtz principle in human perception. According to a basic principle of perception due to Helmholtz [2], an observed geometric structure is perceptually meaningful if it has a very low probability to appear in noise. As a common sense statement, this means that “events that could not happen by chance are immediately perceived”. For example, a group of five aligned dots exists in both images in Figure 1, but it can hardly be seen on the left-hand side image. Indeed, such a configuration is not exceptional in view of the total number of dots. In the right-hand image we immediately perceive the alignment as a large deviation from randomness that would be unlikely to happen by chance.

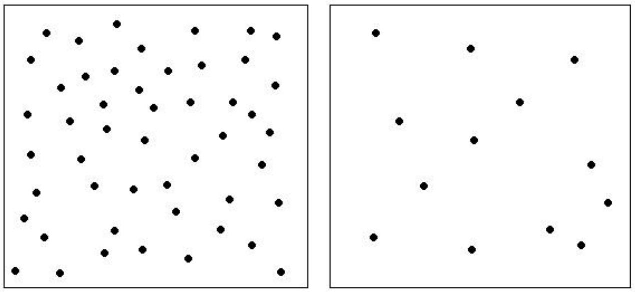


Figure 1. The Helmholtz principle in human perception

In the context of data mining, we shall define the Helmholtz principle as the statement that meaningful features and interesting events appear as large deviations from randomness. In the cases of textual, sequential or unstructured data we derive qualitative measure for such deviations.

Under *unstructured data* we understand data without an explicit *data model*, but with some internal geometrical structure. For example, sets of dots in Figure 1 are not created by a precise data model, but still have important geometrical structures: nearest neighbors, alignments, concentrations in some regions, etc. A good example is textual data where there are natural structures like files, topics, paragraphs, documents etc. Sequential and temporal data also can be divided into natural blocks like days, months or blocks of several sequential events. In this article, we will assume that data comes packaged into objects, i.e. files, documents or containers. We can also have several layers of such structures; for example, in 20Newsgroups all words are packed into 20 containers (news groups), and each group is divided into individual news. We would like to detect some unusual behavior in these data and automatically extract some meaningful events and features. To make our explanation more precise, we shall consider mostly textual data, but our analysis is also applicable to any data that generated by some basic set (words, dots, pair of words, measurements, etc.) and divided into some set of containers (documents, regions, etc.), or classified.

This paper is the first attempt to define document meaning following the human perceptual model. We model document meaning through a set of meaningful keywords, together with their level of meaningfulness.

The current work introduces a new approach to the problem of automatic keywords extraction based on the following intuitive ideas:

- keywords should be responsible for topics in a data stream or corpus of documents, i.e. keywords should be defined not just by documents themselves, but also by the context of other documents in which they lie;
- topics are signaled by "unusual activity", i.e. a new topic emerges with some features rising sharply in their frequency.

For example, in a book on C++ programming language a sharp rise in the frequency of the words "file", "stream", "pointer", "fopen" and "fclose" could be indicative of the book chapter on "File I/O".

These intuitive ideas have been a source for almost all algorithms in Information Retrieval. One example is the familiar TF-IDF method for representing documents [3], [4]. Despite being one of the most successful and well-tested techniques in Information Retrieval, TF-IDF has its origin in heuristics and it does not have a convincing theoretical basis [4].

Rapid change detection is a very active and important area of research. A seminal paper by Jon Kleinberg [5] develops a formal approach for modeling "bursts" using an infinite-state automation. In [5] bursts appear naturally as state transitions.

The current work proposes to model the above mentioned unusual activity by analysis based on the Gestalt theory in Computer Vision (human perception). The idea of the importance of "sharp changes" is very natural in image processing, where edges are responsible for rapid changes and the information content of images. However, not all local sharp changes correspond to edges, as some can be generated by noise. To represent meaningful objects, rapid changes have to appear in some coherent way. In Computer Vision, the Gestalt Theory addresses how local variations combined together to create perceived objects and shapes.

As mention in [6], the Gestalt Theory is a single substantial scientific attempt to develop principles of visual reconstruction. Gestalt is a German word translatable as "whole", "form", "configuration" or "shape". The first rigorous approach to quantify basic principles of Computer Vision is presented in [6]. In the next section, we develop a similar analysis for the problem of automatic keywords extraction.

The paper is organized as follows. In Section II we analyze The Helmholtz Principle in the context of document processing and derive qualitative measures of the meaningfulness of words. In Section III numerical results for State of the Union Addresses from 1790 till 2009 (data set from [7]) are presented and compared with results from [5, Section 4]. We also present some preliminary numerical results for the 20Newsgroups data set [8]. Conclusions and future work are discussed in the Section IV.

II. The Helmholtz Principle and Meaningful Events

We have defined Helmholtz principle as the statement that meaningful features and interesting events appear as large deviations from randomness. Let us now develop a more rigorous approach to this intuitive statement.

First of all, it is not enough to say that interesting structures are those that have low probability. Let us illustrate it by the following example. Suppose one unbiased coin is being tossed 100 times in succession, then *any* 100-sequence of heads (ones) and tails (zeros) can be generated with the same equal probability $(1/2)^{100}$. Whilst both sequences

$$s_1 = 10101\ 11010\ 01001\ \ldots\ 00111\ 01000\ 10010$$

$$s_2 = \underbrace{111111111\ldots111111}_{50 \text{ times}}\ \underbrace{000000000\ldots000000}_{50 \text{ times}}$$

are generated with the same probability, the second output is definitely not expected for an unbiased coin. Thus, low probability of an event does not really indicates its deviation from randomness.

To explain why the second output s_2 is unexpected we should explain what an expected output should be. To do this some global observations (random variables) on the generated sequences are to be considered. This is similar to statistical physics where some macro parameters are observed, but not a particular configuration. For example, let μ be a random variable defined as the difference between number of heads in the first and last 50 flips. The expected value of this random variable (its mean) is equal to zero, which is with high level of accuracy true for s_1. However, for sequence s_2 with 50 heads followed by 50 tails this value is equal to 50 which is very different from the expected value of zero.

Another example can be given by the famous 'Birthday Paradox'. Let us look at a class of 30 students and let us assume that their birthdays are independent and uniformly distributed over the 365 days of the year. We are interested in events that some students have their birthday on the same day. Then the natural random variables will be C_n, $1 \leq n \leq 30$, the number of n-tuples of students in the class having the same birthday. It is not difficult to see that the expectation of the number of pairs of students having the same birthday in a class of 30 is $E(C_2) \approx 1.192$. Similarly, $E(C_3) \approx 0.03047$ and $E(C_4) \approx 5.6 \times 10^{-4}$. This means that 'on the average' we can expect to see 1.192 pairs of students with the same birthday in each class. So, finding two students with the same birthday is not surprising, but having three or even four students with the same birthday would be unusual. If we look in a class with 10 students, then $E(C_2) \approx 0.1232$. This means that having two students with the same birthday in a class of 10 should be considered as an unexpected event.

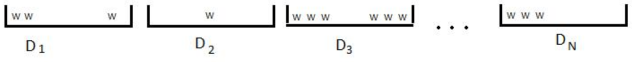
More generally, let Ω be a probability space of all possible outputs. Formally, an output $\omega \in \Omega$ is defined as unexpected with respect to some observation μ, if the value $\mu(\omega)$ is very far from expectation $E(\mu)$ of the random variable μ, i.e. the bigger the difference $|\mu(\omega) - E(\mu)|$ is, the more unexpected outcome ω is. From Markov's inequalities for random variables it can be shown that such outputs ω are indeed very unusual events.

The very important question in such setup is a question of how to select appropriate random variables for given data. The answer can be given by standard mathematical and statistical physics approach. Any structure can be described by its symmetry group. Thus, for any completely unstructured data, any permutation of the data is possible. However, if we want to preserve a structure, then we can only perform structure preserving transformations. For example, if we have a set of documents, then we can not move words between the documents, but can reshuffle words inside each document. In such case, the class of suitable random variables are functions that are invariant under the group of symmetry.

A. Counting Functions

Let us return to the text data mining. Since we defined keywords as words corresponding to a sharp rise in frequency, then our natural measurements should be counting functions of words in documents or parts of documents. Let us first derive the formulas for expected values in the simple and ideal situation of N documents or containers of the same length, where the length of a document is the number of words in the document.

Suppose we are given a set of N documents (or containers) $D_1, \ldots, D_N$ of the same length. Let w be some word (or some observation) that is present inside one or more of these N documents. Assume that the word w appears K times in all N documents and let us collect all of them into one set $S_w = \{w_1, w_2, \ldots, w_K\}$.



Now we would like to answer the following question: *If the word w appears m times in some document, is this an expected or unexpected event*? For example, the word "*the*" usually has a high frequency, but this is not unexpected. On the other hand, the same word "*the*" has much higher frequency in a chapter on definite and indefinite articles in any English grammar book and thus should be detected as unexpected.

Let us denote by C_m a random variable that counts how many times an m-tuple of the elements of S_w appears in the same document. Now we would like to calculate the expected value of the random variable C_m under the assumption that elements from S_w are randomly and independently placed into N containers.

For m different indexes $i_1, i_2, \ldots, i_m$ between 1 and K, i.e. $1 \leq i_1 < i_2 < \ldots < i_m \leq K$, let us introduce a random variable $\chi_{i_1,i_2,\ldots,i_m}$:

$$\begin{cases} 1 & \text{if } w_{i_1}, \ldots, w_{i_m} \text{ are in the same document,} \\ 0 & \text{otherwise.} \end{cases}$$

Then by definition of the function C_m we can see that

$$C_m = \sum_{1 \leq i_1 < i_2 < \ldots < i_m \leq K} \chi_{i_1,i_2,\ldots,i_m},$$

and that the expected value $E(C_m)$ is the sum of expected values of all $\chi_{i_1,i_2,\ldots,i_m}$:

$$E(C_m) = \sum_{1 \leq i_1 < i_2 < \ldots < i_m \leq K} E(\chi_{i_1,i_2,\ldots,i_m}).$$

Since $\chi_{i_1,i_2,\ldots,i_m}$ has only values zero and one, the expected value $E(\chi_{i_1,i_2,\ldots,i_m})$ is equal to the probability that all $w_{i_1}, \ldots, w_{i_m}$ belong to the same document, i.e.

$$E(\chi_{i_1,i_2,\ldots,i_m}) = \frac{1}{N^{m-1}}.$$

From the above identities we can see that

$$E(C_m) = \binom{K}{m} \cdot \frac{1}{N^{m-1}}, \qquad (1)$$

where $\binom{K}{m} = \frac{K!}{m!(K-m)!}$ is a binomial coefficient.

Now we are ready to answer the previous question: *If in some document the word w appears m times and $E(C_m) < 1$, then this is an unexpected event.*

Suppose that the word w appear m or more times in each of several documents. *Is this an expected or or unexpected event*? To answer this question, let us introduce another random variable I_m that counts number of documents with m or more appearances of the word w. It should be stressed that despite some similarity, the random variables C_m and I_m are quite different. For example, C_m can be very large, but I_m is always less or equal N. To calculate the expected value $E(I_m)$ of I_m under an assumption that elements from S_w are randomly and independently placed into N containers let us introduce a random variable $I_{m,i}$, $1 \le i \le N$ with

$$I_{m,i} = \begin{cases} 1 & \text{if } D_i \text{ contains } w \text{ at least } m \text{ times,} \\ 0 & \text{otherwise.} \end{cases}$$

Then by definition

$$I_m = \sum_{i=1}^{N} I_{m,i}.$$

Since $I_{m,i}$ has only values zero and one, the expected value $E(I_{m,i})$ is equal to the probability that at least m elements of the set S_w belong to the document D_i, i.e.

$$E(I_{m,i}) = \sum_{j=m}^{K} \binom{K}{j} \left(\frac{1}{N}\right)^{j} \left(1 - \frac{1}{N}\right)^{K-j}.$$

From the last two identities we have

$$E(I_m) = N \times \sum_{j=m}^{K} \binom{K}{j} \left(\frac{1}{N}\right)^{j} \left(1 - \frac{1}{N}\right)^{K-j}. \qquad (2)$$

We can rewrite (2) as

$$E(I_m) = N \times \mathcal{B}(m, K, p),$$

where $\mathcal{B}(m, K, p) := \sum_{j=m}^{K} \binom{K}{j} p^j (1-p)^{K-j}$ is the *tail of binomial distribution* and $p = 1/N$.

Now, if we have several documents with m or more appearances of the word w and $E(I_m) < 1$, then this is an unexpected event.

Following [6], we will define $E(C_m)$ from (1) as the *number of false alarms* of a m-tuple of the word w and will use notation $NFA_T(m, K, N)$ for the right hand side of (1). The NFA_T of an m-tuple of the word w is the expected number of times such an m-tuple could have arisen just by chance. Similar, we will define $E(I_m)$ from (2) as the number of false alarms of documents with m or more appearances of the word w, and us notation $NFA_D(m, K, N)$ for the right hand side of (2). The NFA_D of an the word w is the expected number of documents with m or more appearances of the word w that could have arisen just by chance.

B. Dictionary of Meaningful Words

Let us now describe how to create a dictionary of meaningful words for our set of documents. We will present algorithms for NFA_T. The similar construction is also applicable to NFA_D.

If we observe that the word w appears m times in the same document, then we define this word as *a meaningful word* if and only if its NFA_T is smaller than 1. In other words, if the event of appearing m times has already happened, but the expected number is less than *one*, we have a meaningful event. The set of all meaningful words in a corpus of documents $D_1, \ldots, D_N$ will be defined as a set of keywords.

Let us now summarize how to generate the set of keywords $KW(D_1, \ldots, D_N)$ of a corpus of N documents $D_1, \ldots, D_N$ of the same or approximately same length:
For all words w from $D_1, \ldots, D_N$

1) Count the number of times K the word w appears in $D_1, \ldots, D_N$.
2) For i from 1 to N
    a) count the number of times m_i the word w appears in the document D_i;
    b) if $m_i \ge 1$ and
    $$NFA_T(m_i, K, N) < 1, \qquad (3)$$
    then add w to the set $KW(D_1, \ldots, D_N)$ and mark w as a meaningful word for D_i.

If the NFA_T is less than ϵ we say that w is *ϵ-meaningful*. We define a set of ϵ-keywords as a set of all words with $NFA_T < \epsilon$, $\epsilon < 1$. Smaller ϵ corresponds to more important words.

In real life examples we can not always have a corpus of N documents $D_1, \ldots, D_N$ of the same length. Let l_i denote the length of the document D_i. We have three strategies for creating a set of keywords in such a case:

- Subdivide the set $D_1, \ldots, D_N$ into several subsets of approximately equal size documents. Perform analysis above for each subset separately.
- "Scale" each document to common length l of the smallest document. More precisely, for any word w we calculate K as $K = \sum_{i=1}^{N} [m_i/l]$, where $[x]$ denotes an integer part of a number x and m_i counts the number of appearances of the word w in a document D_i. For each document D_i we calculate the NFA_T with this K and the new $m_i \leftarrow [m_i/l]$. All words with $NFA_T < 1$ comprise a set of keywords.
- We can "glue" all documents $D_1, \ldots, D_N$ into one big document and perform analysis for one document as will be described below.

In a case of one document or data stream we can divide it into the sequence of disjoint and equal size blocks and perform analysis like for the documents of equal size. Since such a subdivision can cut topics and is not shift invariant, the better way is to work with a "moving window". More precisely, suppose we are given a document D of the size L and B is a block size. We define N as $[L/B]$. For any word w from D and any windows of consecutive B words let m count number of w in this windows and K count number of w in D. If $NFA_T < 1$, then we add w to a set of keywords and say that w is meaningful in these windows. In the case of one big document that has been subdivided into sub-documents or sections, the sizes of such parts are a natural selection for the sizes of windows.

If we want to create a set of ϵ-keywords for one document or for documents of different sizes, we should replace the inequality $NFA_T < 1$ by an inequality $NFA_T < \epsilon$.

C. Estimating of the number of false alarms

In real examples calculating $NFA_T(m, K, N)$ and $NFA_D(m, K, N)$ can be tricky and is not a trivial task. Numbers m, K and N can be very large and NFA_T or NFA_D can be exponentially large or small. Even relatively small changes in m can results in big fluctuations of NFA_T and NFA_D. The correct approach is to work with

$$-\frac{1}{K}\log NFA_T(m, K, N) \tag{4}$$

and

$$-\frac{1}{K}\log NFA_D(m, K, N) \tag{5}$$

In this case the meaningful events can be characterized by $-\frac{1}{K}\log NFA_T(m, K, N) > 0$ or $-\frac{1}{K}\log NFA_D(m, K, N) > 0$.

There are several explanations why we should work with (4) and (5) . The first is pure mathematical: there is a unified format for estimations of (4) and (5) (see [6] for precise statements). For large m, K and N there are several famous estimations for large deviations and asymptotic behavior of (5): law of large numbers, large deviation technique and Central Limit Theorem. In [6, Chapter4, Proposition 4] all such asymptotic estimates are presented in uniform format.

The second explanations why we should work with (4) and (5) can be given by statistical physics of random systems: these quantities represent 'energy per particle' or energy per word in our context. Like in physics where we can compare energy per particle for different systems of different size, there is meaning in comparison of (4) and (5) for different words and documents.

Calculation of (4) usually is not a problem, since NFA_T is a pure product. For (5), there is also a possibility of using the Monte Carlo method by simulating a Bernoulli process with $p = 1/N$, but such calculations are slow for large N and K.

D. On TF-IDF

The TF-IDF weight (term frequency - inverse document frequency) is a weight very often used in information retrieval and text mining. If we are given a collection of documents $D_1, \ldots, D_N$ and a word w appears in L documents $D_{i_1}, \ldots, D_{i_L}$ from the collection, then

$$IDF(w) = \log\left(\frac{N}{L}\right).$$

The TF-IDF weight is just 'redistribution' of IDF among $D_{i_1}, \ldots, D_{i_L}$ according to *term frequency* of w inside of $D_{i_1}, \ldots, D_{i_L}$.

The TF-IDF weight demonstrates remarkable performance in many applications, but the IDF part still remains a mystery. Let us now look at IDF from number of false alarms point of view.

Consider all documents $D_{i_1}, \ldots, D_{i_L}$ containing the word w and combine all of them into one document (*the document about* w) $\widetilde{D} = D_{i_1} + \ldots + D_{i_L}$. For example, if $w =$'cow', then $\widetilde{D}$ is all about 'cow'. We now have a *new collection* of documents (containers): $\widetilde{D}, D_{j_1}, \ldots, D_{j_{N-L}}$, where $D_{j_1}, \ldots, D_{j_{N-L}}$ are documents of the original collection $D_1, \ldots, D_N$ that do not contains the word w. In general,$\widetilde{D}, D_{j_1}, \ldots, D_{j_{N-L}}$ are of different sizes. For this new collection $\widetilde{D}, D_{j_1}, \ldots, D_{j_{N-L}}$ the word w appear only in $\widetilde{D}$, so we should calculate number of false alarms or 'energy' ((4) or (5)) per each appearance of w only for $\widetilde{D}$.

Using an adaptive window size or 'moving window', (4) and (5) become

$$-\frac{1}{K}\log\left(\binom{K}{K}\frac{1}{\widetilde{N}}\right),$$

i.e.

$$\frac{K-1}{K}\cdot\log\widetilde{N}, \quad \text{where} \quad \widetilde{N} = \frac{\sum_{i=1}^{N}|D_i|}{|\widetilde{D}|}. \tag{6}$$

If all documents $D_1, \ldots, D_N$ are of the same size, then (6) becomes

$$\frac{K-1}{K}\cdot IDF(w),$$

and for large K is almost equal to $IDF(w)$. But for the case of documents of different lengths (which is more realistic) our calculation suggest that more appropriate should be *adaptive IDF*:

$$AIDF(w) := \frac{K-1}{K}\cdot\log\frac{\sum_{i=1}^{N}|D_i|}{|\widetilde{D}|}, \tag{7}$$

where K is term count of the word w in all documents, $|\widetilde{D}|$ is the total length of documents containing w and $\sum_{i=1}^{N}|D_i|$ is the total length of all documents in the collection.

III. Experimental Results

In this section we present some numerical results for State of the Union Addresses from 1790 till 2009 (data set from [7]) and for the 20Newsgroups data set [8].

It is important to emphasize that we do not perform any essential pre-processing of documents, such as stop word filtering, lemmatization, part of speech analysis, and others. We simply down-case all words and remove all punctuation characters.

A. *State of the Union Addresses*

The performance of the proposed algorithm was studied on a relatively large corpus of documents. To illustrate the results, following [5], we selected the set of all U.S. Presidential State of the Union Addresses, 1790-2009 [7]. This is a very rich data set that can be viewed as a corpus of documents, as a data stream with natural timestamps, or as one big document with many sections.

For the first experiment, the data is analyzed as a collection of $N = 219$ individual addresses. The number of words in these documents vary dramatically, as shown in Figure 2 by the solid line.

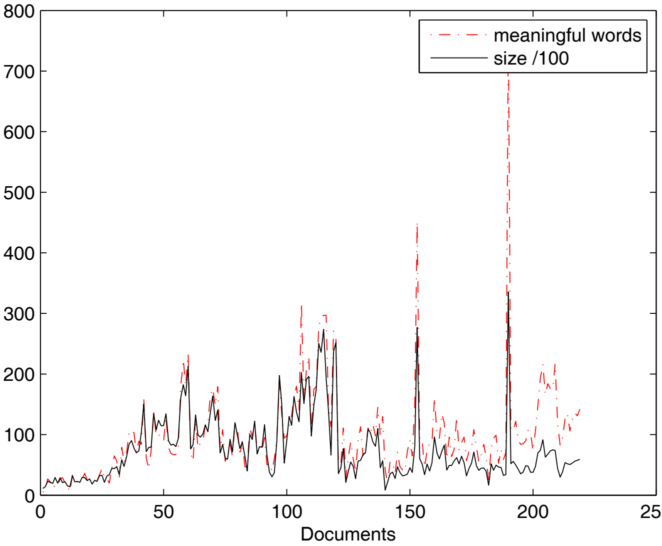


Figure 2. Document lengths in hundreds of words is shown by the solid line; the document average length is equal to 7602.4 and the sample deviation is 5499.7.

As expected, the extraction of meaningful or $\epsilon-$meaningful words using formula (3) from the corpus of different length documents performs well for the near-average length documents. The manual examination of the extracted keywords reveals that

- all stop words have disappeared;
- meaningful words relate to/define the corresponding document topic very well;
- the ten most meaningful words with the smallest NFA follow historical events in union addresses.

For example, five of the most meaningful words extracted from the speeches of the current and former presidents are

Obama, 2009: lending, know, why, plan, restart;
Bush, 2008: iraq, empower, alqaeda, terrorists, extremists;
Clinton, 1993: jobs, deficit, investment, plan, care.

However, the results for the document outliers are not satisfactory. Only a few meaningful words or none are extracted for the small documents. Almost all words are extracted as meaningful for the very large documents. In documents with size more than 19K words even the classical stop word "the" was identified as meaningful.

To address the problem of the variable document length different strategies were applied to the set of all Union Addresses: *moving window*, *scaling to average* and *adapting window size* described in Section II. The results are dramatically improved for outliers in all cases. The best results from our point of view are achieved using an *adaptive window size* for each document, i.e. we calculate (3) for each document with the same K and m_i but with $N = L/|D_i|$ with L being the total size of all documents and $|D_i|$ is the size of the document D_i. The numbers of meaningful words ($\epsilon = 1$) extracted for the corresponding documents are shown by the dashed line in Figure 2. A remarkable agreement with document sizes is observed.

Our results are consistent with the existing classical algorithm [5]. For example, using a moving window approach, the most meaningful words extracted for The Great Depression period from 1929 till 1933 are: "loan", "stabilize", "reorganization", "banks", "relief" and "democracy", whilst the most important words extracted by [5] are "relief", "depression", "recovery", "banks" and "democracy".

Let us now look at the famous Zipf's law for natural languages. Zipf's law states that given some corpus of documents, the frequency of any word is inversely proportional to some power γ of its rank in the frequency table, i.e. frequency(rank)≈const/rank$^{\gamma}$. Zipf's law is mostly easily observed by plotting the data on a *log-log* graph, with the axes being log(rank order) and log(frequency). The data conform to Zipf's law to extend the plot is linear. Usually Zipf's law is valid for the upper portion of the log-log curve and not valid for the tail.

For all words in the Presidential State of the Union Address we plot rank of a word and the total number of the word's occurrences in log-log coordinates, as shown in Figure 3.

Let us look into Zipf's law for only the meaningful words of this corpus ($\epsilon = 1$). We plot the rank of a meaningful word and the total number of the word's occurrences in log-log coordinates, as shown in Figure 4. We still can observe Zipf's law, although the curve becomes smoother and the power γ becomes smaller.

If we increase the level of meaningfulness (i.e. decrease the ϵ), then the curve becomes even smoother and conforms to Zipf's law with smaller and smaller γ. This is exactly what we should expect from good feature extraction and dimension reduction: to decrease the number of features and to decorrelate data.

For two sets S_1 and S_2 let us use as a measure of their similarity the number of common elements divided by the number of elements in their union: $W(S_1, S_2) = |S_1 \bigcap S_2|/|S_1 \bigcup S_2|$. After extracting meaningful words we can look into similarity of the Union Addresses by calculating similarity W for their sets of keywords. Then, for

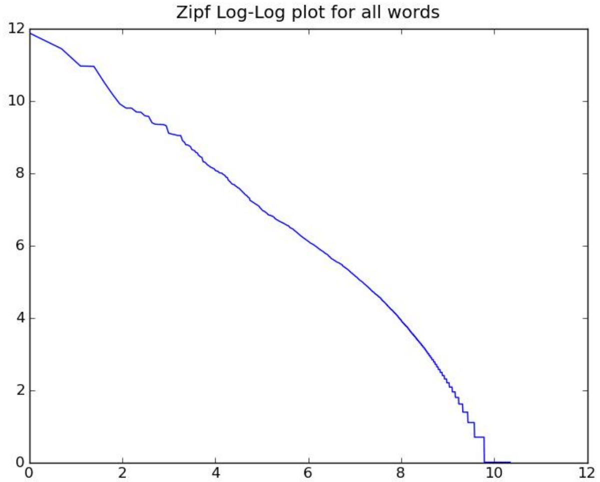


Figure 3.

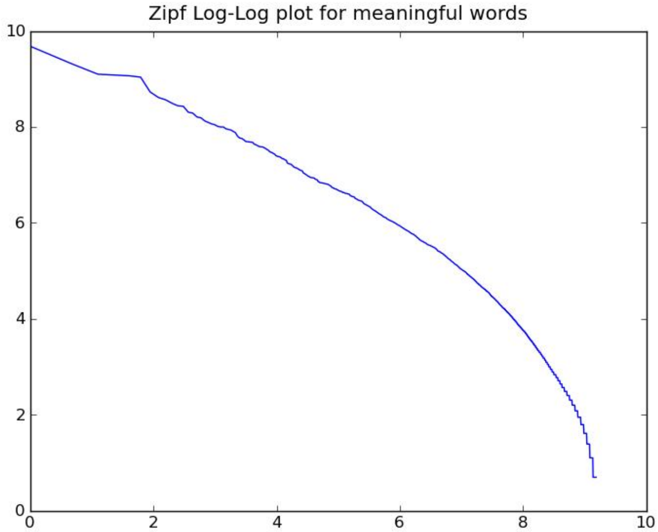


Figure 4.

example, the Barack Obama, 2009 speech is most similar to the George H.W. Bush, 1992 speech with the similarity $W \approx 0.132$ and the following meaningful words in common: set(['everyone', 'tax', 'tonight', 'i'm', 'down', 'taxpayer', 'reform', 'health', 'you', 'tell', 'economy', 'jobs', 'get', 'plan', 'put', 'wont', 'short-term', 'long-term', 'times', 'chamber', 'asked', 'know']).

The George W. Bush, 2008 speech is mostly similar to his 2006 speech (which is very reasonable) with the similarity $W \approx 0.16$ and the following meaningful words in common: set(['terrorists', 'lebanon', 'al-qaeda', 'fellow', 'tonight', 'americans', 'technology', 'enemies', 'terrorist', 'palestinian', 'fight', 'iraqi', 'iraq', 'terror', 'we', 'iran', 'america', 'attacks', 'iraqis', 'coalition', 'fighting', 'compete']).

From all the Presidential State of the Union Addresses, the most similar are William J. Clinton 1997 speech and 1998 speech. Their similarity is $W \approx 0.220339$ and the following meaningful words in common: set(['help', 'family', 'century', 'move', 'community', 'tonight', 'schools', 'finish', 'college', 'welfare', 'go', 'families', 'education', 'children', 'lifetime', 'row', 'chemical', '21st', 'thank', 'workers', 'off', 'environment', 'start', 'lets', 'nato', 'build', 'internet', 'parents', 'you', 'bipartisan', 'pass', 'across', 'do', 'we', 'global', 'jobs', 'students', 'thousand', 'scientists', 'job', 'leadership', 'every', 'know', 'child', 'communities', 'dont', 'america', 'lady', 'cancer', 'worlds', 'school', 'join', 'vice', 'challenge', 'proud', 'ask', 'together', 'keep', 'balanced', 'chamber', 'teachers', 'lose', 'americans', 'medical', 'first']).

B. 20Newsgroups

In this subsection of the article some numerical results for the 20 Newsgroup data set [8] will be presented. This data set consists of 20000 messages taken from 20Newsgroups. Each group contains one thousand Usenet articles. Approximately 4% of the articles are cross-posted. Our only preprocessing was removing words with length ≤ 2. For defining meaningful words we use NFA_T and consider each group as separate container. In Figure 5, group lengths (total number of words) in tens of words is shown by the blue line and the number of different words in each group is shown by the green line. The highest peak in group lengths corresponds to the group 'talk.politics.mideast', and the highest peak in the number of different words corresponds to the group 'comp.os.ms-windows.misc'.

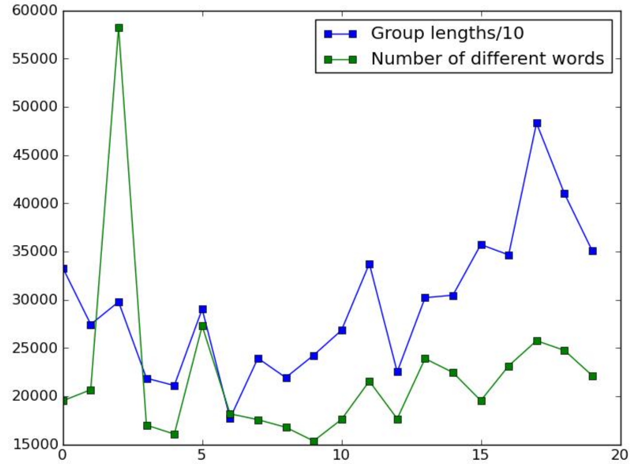


Figure 5.

After creating meaningful words for each group based on NFA_T with $\epsilon = 1$ and removing non-meaningful words from each group, the new group lengths (total number of meaningful words) in tens of words is shown by the blue line in Figure 6. The number of different meaningful words in each group is shown by the green line on the same Figure 6.

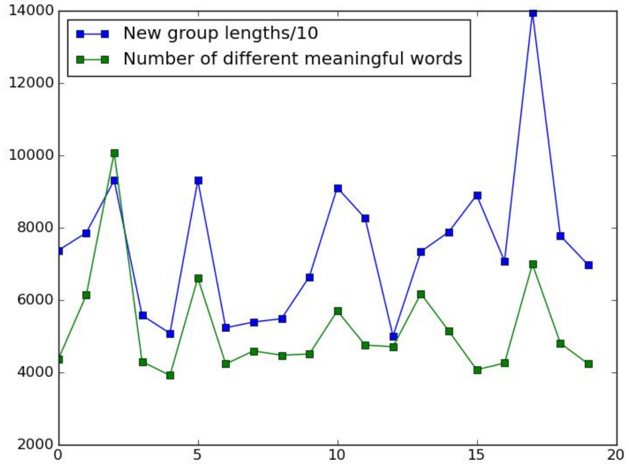


Figure 6.

Let us now look into Zipf's law for 20Newsgroups. We plot the rank of a word and the total number of the word's occurrences in log-log coordinates, as shown in Figure 7, and we also plot the rank of a meaningful word and the total number of the word's occurrences in log-log coordinates, as shown in Figure 8. As we can see, meaningful words also follow Zipf's law very closely.

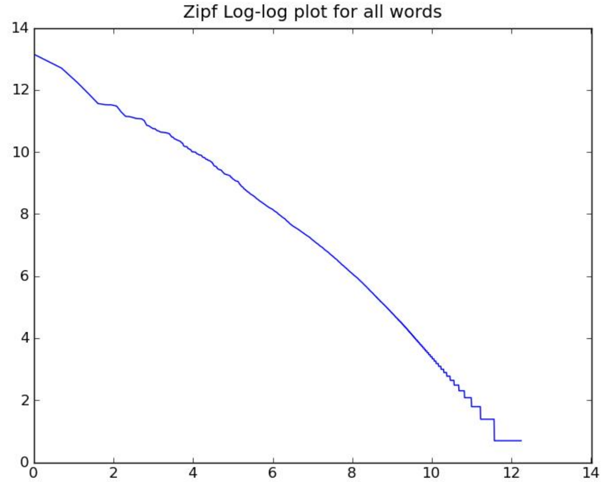


Figure 7.

Similar to the State of the Union Addresses, let us calculate the similarity of groups by calculating W for the corresponding sets of meaningful words. We will index the groups by integer $i = 0, \ldots, 19$ and denote the ith group by Gr[i]; for example, Gr[3] = 'comp.sys.ibm.pc.hardware', as shown on the Table I. The similarity matrix W is a 20×20-matrix and is too big to reproduce in the article. So, we show in the Table I most similar and most non-similar groups for each group, together with the corresponding measure of similarity W. For example, the group ' comp.windows.x' (index=5) is most similar to the group Gr[1]=' comp.graphics' with similarity=0.038, and most non-similar with the group Gr[19]='talk.religion.misc' with similarity=0.0012. As we can see, our feature extraction approach produces very natural measures of similarity for the 20Newsgroups.

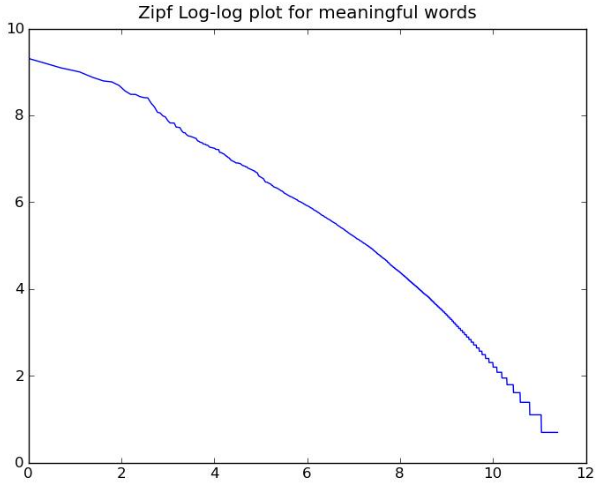


Figure 8.

Let us now investigate how sets of meaningful words change with the number of articles inside groups. Let us create so called mini-20Newsgroups by selecting randomly 10% of articles in each group.In the mini-20Newsgroups there are 100 articles in each group.We have used for our numerical experiments the mini-20Newsgroups from [8]. After performing meaningful words extraction from the mini-20Newsgroup with NFA_T and $\epsilon = 1$, let us plot together the number of meaningful words in each group of original 20Newsgroups, the number of meaningful words in each group of mini-20Newsgroup and the number of common meaningful words for these two data set. The results are shown in Figure 9.

As we can see, a large proportion of meaningful words survives when we ten times increase the number of articles, i.e. when we go from mini to full 20Newsgroup data: red and green lines are remarkably coherent.

Let us now check how all these meaningful words perform in classification tasks. We would like to have a classifier for finding appropriate newsgroup for a new message. Using 10% of news as training set we have created 20 sets of meaningful words, $MW[i]$, $i = 0, \ldots, 19$. Let us introduce the simplest possible classifier C from messages to the

Table I
AN EXAMPLE OF GROUP SIMILARITIES

Index	News Groups	Highest Similarity	Lowest Similarity
0	alt.atheism	Gr[19], 0.12	Gr[2], 0.0022
1	comp.graphics	Gr[5], 0.038	Gr[15], 0.0023
2	comp.os.ms-windows.misc	Gr[3], 0.0197	Gr[15], 0.0023
3	comp.sys.ibm.pc.hardware	Gr[4], 0.041	Gr[17], 0.0024
4	comp.sys.mac.hardware	Gr[3], 0.041	Gr[17], 0.0023
5	comp.windows.x	Gr[1], 0.038	Gr[19], 0.0012
6	misc.forsale	Gr[12], 0.03	Gr[0], 0.0024
7	rec.autos	Gr[8], 0.035	Gr[15], 0.0025
8	rec.motorcycles	Gr[7], 0.035	Gr[2], 0.0033
9	rec.sport.baseball	Gr[10], 0.036	Gr[19], 0.0043
10	rec.sport.hockey	Gr[9], 0.036	Gr[15], 0.0028
11	sci.crypt	Gr[16], 0.016	Gr[2], 0.0025
12	sci.electronics	Gr[6], 0.030	Gr[17], 0.0028
13	sci.med	Gr[12], 0.012	Gr[2], 0.0035
14	sci.space	Gr[12], 0.016	Gr[2], 0.0045
15	soc.religion.christian	Gr[19], 0.044	Gr[2], 0.0014
16	talk.politics.guns	Gr[18], 0.042	Gr[2], 0.0021
17	talk.politics.mideast	Gr[18], 0.022	Gr[2], 0.0018
18	talk.politics.misc	Gr[19], 0.043	Gr[5], 0.0017
19	talk.religion.misc	Gr[0], 0.120	Gr[5], 0.0012

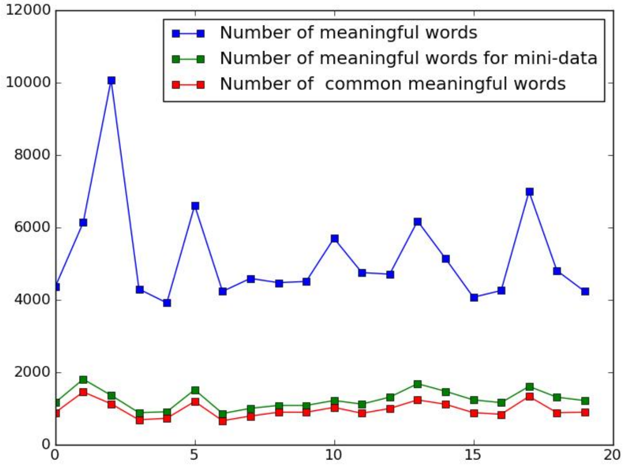


Figure 9.

set of 20Newsgroups. For a message M let us denote by $set(M)$ the set of all different words in M. Then $C(M)$ is a group with largest number of words in $set(M) \bigcap MW[i]$. If there are several groups with the same largest number of words in $set(M) \bigcap MW[i]$, then we select as $C(M)$ a group with smallest index. In the case when all intersections $set(M) \bigcap MW[i]$ are empty, we will mark a message M as 'unclassifiable'.

The results of applying this classifier to the remaining 90% of 20Newsgroups can be represented by the *classification confusion matrix* CCM (Figure 10). For calculating this matrix we used 18000 messages from 20Newsgroups excluding the training set.

CCM is a 20×20 integer value matrix with $CCM(i, j)$ is the number of messages from ith group classified into jth group. For ideal classifier CCM is a diagonal matrix.

$$\begin{array}{cccccccccccccccccccc}
639 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 6 & 2 & 88 & 0 & 5 & 3 & 150 \\
1 & 743 & 10 & 27 & 8 & 78 & 15 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 3 & 1 & 1 & 0 & 2 & 0 \\
0 & 159 & 395 & 75 & 12 & 231 & 18 & 4 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\
0 & 51 & 10 & 702 & 69 & 38 & 21 & 1 & 0 & 0 & 1 & 1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 62 & 2 & 113 & 665 & 19 & 30 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
2 & 139 & 7 & 12 & 1 & 731 & 1 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
1 & 28 & 1 & 30 & 1 & 8 & 795 & 5 & 0 & 4 & 5 & 1 & 11 & 3 & 5 & 1 & 0 & 1 & 0 & 0 \\
0 & 12 & 0 & 8 & 6 & 4 & 81 & 710 & 22 & 11 & 17 & 0 & 13 & 1 & 5 & 0 & 5 & 1 & 3 & 1 \\
0 & 8 & 0 & 3 & 2 & 4 & 30 & 20 & 806 & 3 & 7 & 0 & 6 & 0 & 3 & 4 & 1 & 1 & 2 & 0 \\
1 & 2 & 0 & 3 & 1 & 0 & 12 & 3 & 2 & 833 & 41 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 4 & 2 & 0 & 15 & 874 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
3 & 60 & 0 & 1 & 0 & 8 & 5 & 0 & 1 & 0 & 0 & 813 & 0 & 2 & 2 & 2 & 2 & 0 & 1 & 0 \\
1 & 76 & 2 & 42 & 17 & 14 & 75 & 18 & 1 & 3 & 5 & 12 & 600 & 15 & 16 & 0 & 2 & 1 & 0 & 0 \\
1 & 29 & 0 & 3 & 0 & 8 & 24 & 4 & 6 & 4 & 9 & 5 & 7 & 780 & 10 & 1 & 2 & 1 & 6 & 0 \\
0 & 32 & 0 & 2 & 0 & 3 & 21 & 3 & 2 & 2 & 4 & 4 & 6 & 9 & 803 & 3 & 0 & 1 & 4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 890 & 0 & 1 & 0 & 0 \\
1 & 6 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 2 & 5 & 1 & 2 & 1 & 6 & 777 & 12 & 59 & 25 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 13 & 2 & 857 & 20 & 0 \\
3 & 2 & 0 & 7 & 0 & 1 & 8 & 1 & 1 & 5 & 3 & 2 & 0 & 8 & 12 & 28 & 80 & 73 & 621 & 45 \\
113 & 4 & 0 & 8 & 0 & 1 & 6 & 1 & 0 & 1 & 1 & 0 & 0 & 4 & 3 & 174 & 39 & 9 & 88 & 448
\end{array}$$

Figure 10. Classification confusion matrix, where $CCM(i, j)$ is the number of messages from the i^{th} group classified into the j^{th} group

REMARK: In each row of the CCM, the sum of its elements is equal to 900, which is the number of messages in each group. The exception is the row corresponding to the group "soc.religion.christian", where the sum is equal to 897, because 3 messages from this group remained unclassified, their intersection with the set of meaningful words in each group was empty.

It is also useful to check the classifier performance on the training set itself to validate our approach for selecting meaningful words. The classification confusion matrix for the training set only is shown in Figure 11.

We now calculate the precision, recall and accuracy of our classifier for each of 20 groups.

Precision for ith group is defined as

$$P(i) = \frac{CCM(i,i)}{\sum_j CCM(i,j)}.$$

Recall for ith group is defined as

$$R(i) = \frac{CCM(i,i)}{\sum_j CCM(j,i)}.$$

Accuracy for ith group is defined as harmonic mean of precision and recall

$$A(i) = \frac{2P(i)R(i)}{P(i) + R(i)}.$$

92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7
0	99	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	92	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	98	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	98	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	99	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	99	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
0	0	0	1	0	1	2	0	0	0	0	0	96	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	0	98	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	97	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	2	91	3
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	4	90

Figure 11. Classification confusion matrix for the training set with data presented as in Figure 10

Table II
PRECISION, RECALL AND ACCURACY

News Groups	Precision	Recall	Accuracy
alt.atheism	0.71	0.8331	0.7667
comp.graphics	0.8256	0.5251	0.6419
comp.os.ms-windows.misc	0.4389	0.9229	0.5949
comp.sys.ibm.pc.hardware	0.78	0.6756	0.7241
comp.sys.mac.hardware	0.7389	0.8504	0.7907
comp.windows.x	0.8122	0.6362	0.7135
misc.forsale	0.8833	0.6925	0.7764
rec.autos	0.7889	0.9103	0.8452
rec.motorcycles	0.8956	0.9516	0.9227
rec.sport.baseball	0.9256	0.9423	0.9339
rec.sport.hockey	0.9711	0.9001	0.9343
sci.crypt	0.9033	0.961	0.9312
sci.electronics	0.6667	0.9091	0.7692
sci.med	0.8667	0.9319	0.8981
sci.space	0.8922	0.9209	0.9063
soc.religion.christian	0.9922	0.7325	0.8428
talk.politics.guns	0.8633	0.852	0.8576
talk.politics.mideast	0.9522	0.8899	0.92
talk.politics.misc	0.69	0.7657	0.7259
talk.religion.misc	0.4977	0.6677	0.5703

The results for the precision, recall and accuracy of our classifier for each of 20 groups is shown in the Table II.

As we can see from the Table II, this simple classifier performs impressively well for the most of news groups, thus illustrating the success of NFA_T for selecting meaningful features. The smallest accuracy of around 57% has been observed for the group ‘talk.religion.misc’. From the classification confusion matrix CCM (Figure 10), we can see that many articles from ‘talk.religion.misc’ have been classified as belonging to ‘alt.atheism’, ‘ soc.religion.christian’ or ‘talk.politics.misc’ groups.

IV. CONCLUSION AND FUTURE WORK

In this article, the problem of automatic keyword and feature extraction in unstructured data is investigated using image processing ideas and especially the Helmholtz principle. We define a new measure of keywords meaningfulness with good performance on different types of documents. We expect that our approach may not only establish fruitful connections between the fields of Computer Vision, Image Processing and Information Retrieval, but may also assist with the deeper understanding of existing algorithms like TF-IDF.

In TF-IDF it is preferable to create a stop word list, and remove the stop word before computing the vector representation [1]. In our approach, the stop words are removed automatically. It would be very interesting to study the vector model for text mining based with $-\log(\mathrm{NFA})$ as a weighting function. Even the simplest classifier based on meaningful events performs well.

One of the main objectives in [6] is to develop parameter free edge detections based on *maximal meaningfulness*. Similarly, algorithms in data mining should have as few parameters as possible – ideally none. Developing a similar approach to the keyword and feature extraction, i.e. defining the maximal time or space interval for a word to stay meaningful, is an exciting and important problem. It would also be interesting to understand the relationship between the NFA and [5].

REFERENCES

[1] A. N. Srivastava and M. Sahami, Eds., *Text Mining: classification, clustering, and applications*. CRC Press, 2009.

[2] D. Lowe, *Perceptual Organization and Visual Recognition*. Amsterdam: Kluwer Academic Publishers, 1985.

[3] K. Spärck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[4] S. Robertson, “Understanding inverse document frequency: On theoretical arguments for idf,” *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[5] J. Kleinberg, “Bursty and hierarchical structure in streams,” in *Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2002.

[6] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, ser. Interdisciplinary Applied Mathematics. Springer, 2008, vol. 34.

[7] (2009) Union Adresses from 1790 till 2009. [Online]. Available: http://stateoftheunion.onetwothree.net/

[8] (1999) 20 Newsgroups Data Set. [Online]. Available: http://kdd.ics.uci.edu/databases/20newsgroups/