Universiteit Leiden

ICT in Business

Ranking of Multi-Word Terms

Name: Ricardo R.M. Blikman
Student-no: s1184164

Internal report number: 2012-11

Date: 07/03/2013

1st supervisor: Prof. Dr. J.N. Kok

2nd supervisor: Dr. P.W.H. van der Putten

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Text mining, also known as text data mining or knowledge discovery in textual sources, is the process of extracting interesting and non-trivial patterns or knowledge from textual sources. One subtask is to provide an overview of frequently used terms. Terms are groups of one or more words in a specific order. By giving an overview of most frequently used terms in document collections we hope to obtain knowledge about its contents. Simply counting terms, relative to the source or not, however, can give us a wrong view on its content. The more words in a term gives a term more context but the process of putting terms into context has several challenges, one of the major ones is to rank them. Single word terms like house, car, movie do not tell us much and can be used in different contexts leaving us often clueless on what a term is used for and its relation to other terms, thus; simply counting low word count terms on occurrence leads to loss of knowledge. In this research we present a new approach to rank them by relevance in a fairly simple way. It is possible to rank terms that consist out of different word lengths without the regular problems that occur when using solely the count of appearance of a term. This approach can be used to extract multi-word terms from collections of various textual sources and gives insight into its content by putting the extracted terms into context. The method does not need a dictionary and is configurable, meaning; it can be based on any text mining algorithm and stop list.

TABLE OF CONTENTS

1. Introduction

In the last decades text mining is being used extensively for various purposes e.g. trend spotting, search engines and mining medical documents for new relations between entities. A lot of work has been done in this area and various methods have been created covering a statistical approach, linguistic approach or both (Frantzi et al., 1998).

There is a need for rapid processing of big quantities of information into knowledge. The challenge of processing an abundance of information into knowledge is a shortage of human processing capacity. The necessity to analyze large amounts of data in a pro-active / predictive manner and unveil complex patterns that are embedded in data sets can exceed human comprehension (intellectual grasp).

Imagine a person is missing and that the only information is inside 1,000 saved MSN conversations. To figure out a motive or to look for clues one must understand someone's way of life. What is important to that person (terms)? What does this person talk about (trends)? How does he communicate with peers (slang / unknown words / different languages)? Of course one could simply count terms and make a list based on statistics, but what does it say? What is the context in which a word is used? What is the relevancy of the words? Can a pattern be found? Multi word terms contain more context then single words, making them often more relevant.

There are several known ways to discover terms, both single and multi-word, out of a corpus and there are several methods to uncover them, statistical, linguistic or both. Words can also be put into context by the use of dictionaries and / or maps (Palakal et al., 2002) that put certain combinations of terms into the proper context. Another method is a language model approach to key phrase extraction (Tomokiyo & Hurst. 2003) which uses language models based on a background corpus to predict new terms out of a foreground corpus.

One of the problems encountered when trying to combine single and multi-term words is that single words tend to appear more frequent than multi-term words. When trying to discover trends and mix both single and multi-words it obviously result in most single term words ranking higher then multi-term words. There is no good way yet to rank multi word terms and that capture the meaning, such that we will get an indication what the content of a large collection of documents is about.

The issue this thesis is dealing with is to rank terms that consist out of a variable amount of words by relevance instead of frequency of appearance. We will introduce a method for this purpose called the B Ranking-Method. It can be used for text mining unknown text sources containing both known/unknown, single or multi-term words and put them in a proper context to provide more insight regarding its contents and improve the knowledge that is extracted from the source data.

Making lists, finding new terms and ranking is not new (Tomokiyo & Hurst, 2003). The combination of two concepts and put terms in a context, based on various inputs and algorithms is. The focus of this research lies upon putting terms into context by combining single and multi-word terms and rank them.

The research question of this thesis is:

Can we weigh the relevance of single and multi-word terms and combine them into one list?

In order to satisfy the main research question the following sub questions are defined:

1. Can we extract multi-word terms out of small text document collections?
2. Can we make sense out of multi-word terms without using dictionaries?
3. Can current methods be applied to large collections of small text documents?

We claim that by scoring multi-termed words using currently used methods, e.g. the C-value/NC-value method (Frantzi et al., 1998) or use a language model approach (Tomokiyo & Hurst, 2003), and rank them based on a newly designed algorithm it is possible to rank multi-termed words properly and give us a better understanding in what a pile of random documents are mostly about. The method should be short, understandable and simple to implement.

We run experiments to test our method on various corpuses. The rest of this master thesis is organized as follows. Section 2 will provide some background on text mining. Section 3 covers text mining scoring algorithms. Section 4 will explain term ranking and its application. In section 5 we will explain the B Ranking-Method. Section 6 describes our experiments, the results and the conclusions based on our experiments. Finally, section 7 is for discussion.

2. Text Mining

How to make sense out of a pile of documents? What are the documents mostly about? Can a trend be revealed? What can we say about the documents without reading all of them? The problem this thesis addresses is how to make sense out of large amounts of data. In short; how can we rapidly process big quantities of information into intelligence? A shortage of human processing capacity requires the necessity to analyze in a pro-active / predictive manner and unveil complex patterns that are embedded in data sets, which exceeds human comprehension.

Text (data) mining or knowledge discovery from textual sources refers to the process of extracting interesting and non-trivial patterns or knowledge from unstructured text fragments or documents. It can be considered as an extension of data mining or knowledge discovery from textual sources like databases or collections of text documents. In order to obtain knowledge from text one must first extract relevant terms from the source data. Terms can be extracted using various term extraction methods in combination with stop list filters and or dictionaries. Terms are groups of one or more words, more words in a term generally provides more context for a term more context but to many words in a term decreases its frequency. This is one of the major challenges in the process of putting terms into context. For more information about text data mining consult Fayyad et al. (1996).

2.1 Terms

Terms, the linguistic representation of concepts, Sager et al. (1980). What do they mean? What can terms tell us? These questions look simple at first but when you give it more thought only more questions appear.

We define a term as a single or set of words. Not all words are useful for text mining, extracting meaning from text, for example, words like ‘a’, ‘the’, ‘I’ are too common and do not provide us information. In text mining we refer to useful words or group of words as terms. Like with “text” is both a word and a term, “text data mining” is a term consisting out of three words and “text processor” is a term with two words, but all these terms are not the same and refer to completely different concepts.

In computer logic where integer 32 bits means a sequence of 32 bits represented as a number with a fixed minimum or maximum length, no matter to which computer you speak. Terms do not. Humans have multiple terms for the same concept, or terms which can mean something completely different or even opposite when placed in another context, also the size of writing or the tone of speech can change its meaning and also each person that evaluate a term can give it another meaning. For instance, when someone writes: *“I do not like the white house.”* What does it tell us? It can refer to someone who is deciding which house to buy and he does not like the house which is painted white, or it can be an extremist who does not like the US government. Both answers can be found logical when put into the proper context and both can also be illogical when they are not.

Why would we give a different meaning to the same term? It is because humans consider context or sometimes no context at all. If we take 1.000 movie reviews from the internet movie database and compute the results will be pretty obvious the top ranked words will be “a”, ”the” ,”I” etc. and “Film”, “Movie” and numbers ranging from 0 to 10. Of course text mining has ways to remove certain words, stop words, so most likely only “Movie” or “Film” would score. So, can this be useful to us? It can be useful under very specific conditions; however, we should focus first on the question: why would we want to do that? We got content based on 1.000 movies so what can we do with that? We could use it to figure out if there is a trend in movies. What are most movies about, what can the movie reviews tell us? If we can discover the trend in movies one can imagine what we could do with this knowledge, however; using the basic way the trend will be: “Film” and ”Movie”. So what could we do? We could filter out words and use work very hard to construct dictionaries and custom stop list to filter out words like “Film” and “Movie” and we have a better result. One of the problems one gets is that you will not find trends which contain these words. For example: “Scary movie” would be filtered. So what else could we do? We can focus on extracting terms instead of words and try to put them in context.

2.2 Text mining approaches for term extraction

In this thesis we focus on term extraction and scoring. There are a number of approaches in the domain of text mining for extracting multi-term words. For an general overview consult: SanJuan et al (2006).

There are several known approaches to term extraction and finding multi-term words. We will not describe each one, instead we describe the underlying method that is being used. The following types of methods exist:

- *Statistical*
  Statistical information derived from word frequency and distribution is used by the machine to compute a relative measure of significance. High-quality information is typically derived from patterns and trends through means such as statistical pattern learning

- *Syntactical*
  Syntactical text mining refers to the addition of one or more words to an existing term as in information retrieval and efficient retrieval of information. We call these operations expansions. Expansions that affect the modifier words are further broken down into left-expansion and insertion. Alternatively, expansions can affect the head word. In this case, we talk of right expansion. In short syntactical text mining discovers words based on grammar.

- *Semantical*
  Relating words / symbols based on distinction between the meanings of words. In text mining it is the process of relating syntactic structures, from the levels of phrases, clauses, sentences and paragraphs to the level of the writing as a whole, to their language-independent meanings. It also involves removing features specific to particular linguistic and cultural contexts

- *Morphological*
  Morphological text mining is based on the patterns of word formation in a particular language, including inflection, derivation, and composition. It refers to number and gender variations in a term and also to spelling variants, for example "house" and "houses". It enables the machine to recognize different appearances of the same term.

- *Terminological*
  Discover and determine the relevance of words based on terminology. Term extraction, term recognition, or glossary extraction, is a subtask of information extraction. The goal of terminology extraction is to automatically extract relevant terms from a given corpus.

- *Hybrid*
  A combination of one or more methods mentioned above.

3. Text Mining Scoring Functions

Text mining scoring functions are used to score terms so they can be weighted, this can be done by a simple count or by using more sophisticated methods that count the frequency of a term compared to the frequency of other terms in other documents. We will take a look at several text mining scoring functions and see how they work. We will focus on the C-value / NC-value (Frantzi et al., 1998) and $tf * idf$ based algorithms. We choose these methods because they are well known and are used by many scientists in the area to score terms. However in our approach any other scoring function can be used.

3.1 TF-IDF

The $tf * idf$ weight (term frequency–inverse document frequency) is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is used as a weighting factor in information retrieval and text mining. The $tf * idf$ value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

The term count $tf(t, d)$ in a document is simply the number of times a term t appears in that document d. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term t within the particular document d.

The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

with $|D|$: cardinality of D (the total number of documents in the corpus), and $|\{d \in D : t \in d\}|$: number of documents where the term t appears. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the formula to $idf(t, D) = \log \frac{|D|}{1+|\{d \in D : t \in d\}|}$. Mathematically the base of the log function does not matter and constitutes a constant multiplicative factor towards the overall result.

Then the $tf * idf$ value is defined by:

$$tf * idf(t, d, D) = tf(t, d) \text{ x } idf(t, D) .$$

A high weight in $tf * idf$ is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than 1, the value of idf(and $td * idf$) is greater than 0. As a term appears in more documents then ratio inside the log approaches 1 and making idf and $td * idf$ approaching 0. If a 1 is added to the denominator, a term that appears in all documents will have negative idf, and a term that occurs in all but one document will have an idf equal to zero.

3.2 C-value / NC-value

The C-value / NC-value (Frantzi et al., 1998) is a hybrid approach that combines a statistical with a linguistic approach. In short, it determines the C-value by combining linguistic and statistical information with the emphasis on the statistical part. The C-value is defined by:

$$C(\alpha) = \begin{cases} log^2|\alpha| f(\alpha) & \alpha \text{ is not nested} \\ log^2|\alpha| (f(\alpha) - \frac{1}{P(T\alpha)} \sum_{b \in T\alpha} f(b)) & otherwise \end{cases}$$

where α is the candidate string, f(.) its frequency of occurrence in the corpus, Tα denotes the set of extracted candidate terms that contain α and $P(T\alpha)$ denotes the number of candidate terms in T. A candidate term can be a term on itself or it can be nested as a word within a multi-word term.

The C-value can be extended to the NC-value, which uses context information for the extraction of multi-term words. It measures the weight of a word the following way:

$$weight(w) = \frac{t(w)}{n}$$

where w is the context word (noun, verb or adjective) to be assigned a weight as a term context word, $t(w)$ is the number of terms the word w appears in and n is the total number of terms considered. The purpose of the denominator n is to express this weight as a probability: the probability that the word w might be a term context word.

The NC-value is defined as follows:

$$\mathrm{NC}-\mathrm{value(a)} = 0.8\ \mathrm{C}-\mathrm{value\ (a)} + 0.2 \sum_{b \in Ca} f_{\alpha}(b) weight(b)$$

where a is the candidate term, Ca is the set of context words of a, $f_{\alpha}(b)$ is the frequency of b as a term context word of a, $weight(b)$ is the weight of b. The constants 0.8 and 0.2 where used during their experiment.

The C-value is based on the frequency of a candidate term – the occurrence of the term in longer candidate terms. The greater the number, the bigger its independence and vice versa. The positive effect of the length of the candidate string is moderated by the application of the logarithm on it. The NC-value is broken up into three stages: One, apply the C-value method to the corpus and make a list based on its C-value; Two, extraction of the context terms and their weights; Three, re-rank the list by incorporating the context information from step two and determine the context factor by calculating the weight of a term based on its appearance as a sub term based on the constants for the C-value 0.8. The constant 0.2 is used in the second part of the formulae for the NC-value as in the experiment conducted by (Frantzi et al., 1998).

For the linguistic part the C-value / NC-value uses:

- Part-of-Speech information from tagging the corpses.
  Part-of-speech tagging is the assignment of a grammatical tag (e.g. noun, adjective, verb, preposition, determiner, etc.) to each word in the corpus. It is needed by the linguistic filter which will only permit specific strings for extraction.

- A linguistics filter.
  The linguistics filter is used to exclude those strings not required for extraction based on a dictionary of predefined strings not required for extraction.

- A stop-list.
  A stop-list is a list of words which are not expected to occur as term words in that domain. It is used to avoid the extraction of strings that are unlikely to be terms, improving the precision of the output list.

The method improves the precision of extracting nested multi-word terms by using more statistical information then the pure frequency of occurrence. It also improves distribution of real terms in a ranking list by placing most non–terms to the bottom. The method has only been tested on a medical corpus that belongs to a specific text type that covers well-structured texts in one language.

4. Terms

4.1 Term extraction

Extracting terms from a text document is a difficult task when one wants to extract multi-word terms. For example look at the following phrase, "The Terminator is an exciting action movie". Looking at the word action or movie both could be valid terms; however we would be more interested in the term "action movie" then "action" or "movie".

We want to find a way to identify multi word terms. There are several methods to tackle this problem.

Static dictionary

One could use a static dictionary, however, such a dictionary is hard to maintain and does not recognize unknown or new multi term words.

NLP parsers

Natural language processing (NLP) is concerned with the interactions between computers and human (natural) languages. The paper (Yakushiji et al., 2000) uses a full NLP parser to extract information from biomedical papers. They use two preprocessors to resolve local ambiguities in sentences to improve efficiency.

Relationships extracted by using NLP tend to be too specific to be extended to new domains without creating new rules for new relationships. We therefore prefere another method.

Hybrid method

The paper (Tomokiyo & Hurst, 2003) proposes a new model. The model is able to extract both directional and hierarchical relationships. It is also able to adapt to different biological problem domains using learning methods. Three steps are taken to identify and tag objects:

1. Use multiple dictionaries to identify known objects.
2. Use Hidden Markov models (HMM) to identify unknown objects based on term suffixes.
3. Use N-Gram models to resolve object name ambiguity

It uses the following formulae:

$$\mathrm{P}(w_1, \ldots, w_n) = \prod_{k=1}^{n} P(w_k \mid w_{k-n+1}^{k-1})$$

where P denotes the probability of a sequence of words $w_1, \ldots, w_n$ the length n.

4.2 Term ranking

The main issue this thesis is facing is: How to rank multi word terms? A simple answer to this question can be: count the terms and make a list based on the count. This answer however will simply not satisfy most needs for term ranking. The reason for this is simple, single word terms tend to occur more often than multi word terms. Imagine that we take 1.000 documents extract single and multi-word terms and then count them. We will probably end up with thousands of terms and obviously the terms that will occur most will be single word terms. E.g. a top 100 list consists almost only of single word terms.

The reason we would like to have a list is because we will get an information overload of terms if we do not properly rank them by relevance. Multi-word terms that occur less than single word terms does not make them less relevant per se. The goal of this thesis is to retrieve a list of relevant multi-word terms from document collections. We will propose a new method that will focus on this aspect of text mining.

5. The B RANKING-METHOD

In this section we propose a method for multi-word term extraction and ranking. The reason we want to propose a new method is we look for a scoring method independent way to determine the context of a term and give us insight in the content of a document collection (the disadvantage of the method is that if the parameters are set too low it is less strict and accepts more terms, on the other hand; when set to strict it will dismiss a lot of multi-words terms). The B Ranking-Method has three steps, is configurable and can be used with any text mining algorithm that relates term frequency to the total amount of terms extracted. The B Ranking-Method can be applied relatively simple. The B Ranking-Method has three steps:

1. Term extraction.
2. Weighing terms (based on a scoring mechanism).
3. Determine the relevance of terms(based on the words and length of a term).

Each step is described in their respective subsection below. An important concept of the B Ranking-Method is that algorithms and weights used are parameters. One could test the same algorithm with different weights and cross compare the results based on the amount of source documents available. It is also possible to train and use a corpus to predict a term, or remove a training corpus and use the source without any reference at all.

5.1 Term extraction

This subsection describes the steps taken in the B Ranking-Method algorithm to merge multi word term lists and re-rank the multi word terms. In the preprocessing phase all text is combined into one source, either in one data file, memory or a database table. The reason is that we do not want to predict a term based on old or non-domain specific documents because it will not help discovering new terms. To assure this the weight of a term is determined against all the terms in the entire collection. In addition the threshold we have set for a term is that a term must consist out of a word with a minimum of two letters. Each word is a term, however, if a multi word term is found, the word elements are removed as a term in lower word term counts unless the term does not reach its term threshold. For example, if the multi word term "action movie" does not appear at least as many times as the threshold, it will be considered as two terms "action" and "movie", if it does appear five or more times it is considered as one multi term word "action movie". If the multi term "action movie" appears more times as a sub term in another multi word term it will be removed as a two word term and considered part of this new multi word term.

There are three parameters to be configured (threshold):

1. Minimum word length for a term (TL).
2. Minimal occurrence of a term (O).
3. Maximum amount of words in a multi word term (MT).

A term t for the B Ranking-Method is called valid when the word length of t is larger than TL.

In the first cycle, all single word terms which are valid are found and registered. In the second cycle all two word terms are evaluated and if a multi word term exceeds the threshold, each word in the multi word term is removed as a single term and registered as a two word term. This continues in the third, fourth, fifth cycle etc. until the maximum length for a multi word term (MT) is reached or there is no valid multi word term found for a specific length. Since we start at terms consisting of one word and build up the list of single and multi-word terms in a linear way, we can conclude that when there are no valid terms of a specific word length found, terms which consist out of more words will not be found either. However; in practice it makes sense to set a maximum amount of words in a multi word- term.

5.2 Weighing a term

After terms are extracted we have several lists of terms. Each list is based on the number of words a valid term has. As mentioned earlier, one cannot just simple compare single word terms with multi word terms based upon frequency. To tackle this problem the terms registered in lists that are being used by the B Ranking-Method two values must be registered:

1. The occurrence of a term.
2. The weight of a term.

Getting the occurrence of a term is simply gained by counting the amount of times it appears within the corpus. One has to keep in mind that one does not want to count sub terms.

Weighing a term is complex and the B Ranking-Method allows using any algorithm for this task. We will make use of the binomial log likelihood algorithm (Dunning, 1993). The log likelihood statistic is computed by a function, whose program is given in appendix A of this document.

However; any method of weighing can be used. What is important is that terms are weighted against the total amount of words. One cannot simply count term occurrences and not weigh them against the total amounts of words. If a term is not weighted against the total amount of terms the B Ranking-Method will not succeed in properly ranking terms and lead to random results based on the specific situation. The reason is that the occurrence of a term is relative, for example, if the term “Leiden University” appears 1.000 times within 1.000 documents it can be a relevant term (depending on the amount other terms occur) but if the data consists out of the entire internet 1.000 occurrences is not relevant at all.

As the amount of data increases more terms appear and occurrence compared to relevance will change. If this rule is not taken into consideration it will eventually lead to ranking lists which have single word terms listed in the top because they appear more often. When one not weigh a term properly against the amount of data, the exact opposite is also true. When the amount of data decreases, more multi word terms will appear at the top of a list. If the amount of data is too small chances are multi word terms will not be discovered at all simply because the occurrence of multi word terms will most likely stay beneath the minimum threshold value for multi word terms.

5.3 Determine the relevance of a term

When we have obtained a list of single and multi-word terms with weights it is still not useable. The heaviest weighted terms will still be the ones which occur more which in turn are most likely single word terms. Even though the weighing method used is usable for terms which consist out of the same amount of words, it is not usable when comparing terms that do not contain the same amount of words. To solve this problem the B Ranking-Method uses the following formulae:

$$Bval(t) = Tl(t) * Tf(t) * Tw(t)$$

where $Tl(t)$ denotes the number of words in a term t, $Tf(t)$ denotes the frequency of a term t and $Tw(t)$ denotes the weight of a term t. As a side effect the Bval might assign terms a Bval of "0". The terms that scored "0" provides us with an interesting view on words / terms which cannot be evaluated for a variety of reasons. The information the B Ranking-Method produces for these words / terms which cannot be weighted and score "0" give insight for optimization of either the algorithm used, the initial weights of word terms, changes in stop word lists or errors in the datasets used.

The reason why the B Ranking-Method uses this formulae is because multi word terms tend to be more relevant than single term words because they tend to provide more context e.g. the single word term "movie" tend to appear more frequent than multi word term "action movie" while the context of the single word term "movie" provides less context than the multi word term "action movie". When one purely looks at the frequency of a term single word terms also tend to populate the top results list because they tend to appear more frequent. If one would mine a corpus of documents about movies the single word term "movie" would most likely appear more frequent then the multi word terms "action movie" or "horror movie" whilst the multi term words could tell us more about the content of the corpus they would be ranked very low or maybe even outside the top term list and the single word term would be ranked very high. If you look at it from a statistical point of view this is correct but when we are mining text for information this is not practical. The reason why we do this is the same reason stop lists are used, some terms are to general and provide little or no information or context whatsoever. The B Ranking-Method deals with this issue by increasing the relevance of longer and multi term words.

5.4 Configuration

The B Ranking-Method is configurable; the main reason is that different amounts of data require different approaches, but also important is the amount of resources and time needed to compute the results. If there is a lot of data, a high threshold for terms can be set, this could be automated; Also it can be interesting to use a different method for weighing a term. The implementation of the B Ranking-Method can depend on the situation, the resources available, time and underlying problem. It can be interesting to use two different settings of parameters used by the B Ranking-Method and compare the results. Keep in mind that the parameters of the minimum frequency a term have a direct relation with the amount of data you use it on. Multi-word terms containing a lot of words can be weighted to heavy when the corpus used contains too few terms.

For the experiments a custom implementation based on "A Language Model Approach to Keyphrase Extraction" (Tomokiyo & Hurst, 2003) is used as a scoring method. Terms are collected inside language models which are used to calculate a score based on the occurrence of a term compared to the occurrence of other terms, and the total of all terms in the corpus. Scoring $S(t)$ within the models is implemented the following way:

$$S(t) = \frac{O(t) + 0.5}{Oall + Mall * 0.5}$$

where $O(t)$ denotes the occurrence of a term t, $Oall$ is the number of terms in the language model and $Mall$ the count of all different terms in the model.

6. Experiments

In this section we discuss the experiments we have conducted with the B Ranking-Method on a corpus containing 1.000 reviews from the internet movie database (http://stuff.mit.edu/afs/athena/course/6/6.863/share/data/corpora/movie_reviews/pos/). To perform the computing and visualize the results prototype software has been written. The results of this experiment give insight in what the data is about and what are the main topics / buzzwords / trends in this document collection. The term extraction component has no background corpus.

Each run of the experiment is conducted in three steps after all text fragments are preprocessed into one source:

1. Determine valid terms by setting a minimum term length, a minimal term occurrence, a maximum amount of words for a valid term and define a stop list. This sets our strictness to the terms we are interested in. For example, if a term occurs only once or twice compared to 10.000 other terms in the source it is not relevant for ranking.
2. Select a text mining algorithm for weighing terms. In this case the binominal log likelihood algorithm.
3. Evaluate the results based on qualitative tests in the source documents.

If the results of the experiment are not good in the sense that, the configuration set in steps one and two will be modified and the experiment is repeated until we conclude the method does not work or until we can complete step three. With the proper configuration we can piece together what the data is telling us about its content. To verify the results we will read 50 reviews (5% of the total text), selected random, and judge if the information is in line with the trend.

The data that is used for this experiment are one thousand random movie descriptions from the internet movie database. The results of applying the B Ranking-Method must give us a top 100 score of multi-word terms and give us insight in what the main trends / movie genres / actors / buzzwords – phrases are. The data is selected randomly and we do not have any prior knowledge about its content except it consists out of 1.000 positive movie reviews. The reviews can be downloaded at:
http://stuff.mit.edu/afs/athena/course/6/6.863/share/data/corpora/movie_reviews/pos/.

An example of the data (cv000_29590.txt):

```
films adapted from comic books have had plenty of success , whether they're about superheroes
( batman , superman , spawn ) , or geared toward kids ( casper ) or the arthouse crowd ( ghost
world ) , but there's never really been a comic book like from hell before .
for starters , it was created by alan moore ( and eddie campbell ) , who brought the medium to
a whole new level in the mid '80s with a 12-part series called the watchmen .
to say moore and campbell thoroughly researched the subject of jack the ripper would be like
saying michael jackson is starting to look a little odd .
the book ( or " graphic novel , " if you will ) is over 500 pages long and includes nearly 30
more that consist of nothing but footnotes .
in other words , don't dismiss this film because of its source .
if you can get past the whole comic book thing , you might find another stumbling block in
from hell's directors , albert and allen hughes .
getting the hughes brothers to direct this seems almost as ludicrous as casting carrot top in
, well , anything , but riddle me this : who better to direct a film that's set in the ghetto
and features really violent street crime than the mad geniuses behind menace ii society ?
the ghetto in question is , of course , whitechapel in 1888 london's east end .
it's a filthy , sooty place where the whores ( called " unfortunates " ) are starting to get a
little nervous about this mysterious psychopath who has been carving through their profession
with surgical precision .
when the first stiff turns up , copper peter godley ( robbie coltrane , the world is not
enough ) calls in inspector frederick abberline ( johnny depp , blow ) to crack the case .
abberline , a widower , has prophetic dreams he unsuccessfully tries to quell with copious
```

6.1 Initial Results

Two runs where done, in the first run we configured the minimum word length of a valid term to three letters and as a result many terms had a B-value of 0. It looked imperative a multi word term can consist out of words with a length of two letters because terms like “kung-fu” or “kung fu” are broken into two words by the stop word list we used. After a brief evaluation we decided to reconfigure the variables we had set for the B Ranking-Method and applied it again with the following settings:

Stop list:	Basic English
Custom stop words:	, . - : + * = ; &
Minimum word length for a term:	2
Minimum occurrence of a valid term:	3
Maximum words for a term:	3
Text mining scoring algorithm:	binomial log likelihood algorithm (Dunning, 1993)

The experiment produced the following results:

Rank	Term	Frequency	Score	Count	Lenght	Bval
1	film	3849	-0,008666238	1	4	-133,4254
2	movie	1950	-0,004423084	1	5	-43,1250725
3	time	948	-0,00215919618	1	4	-8,187672
4	story	869	-0,00197996176	1	5	-8,602934
5	character	821	-0,00187100645	1	9	-13,8248663
6	characters	789	-0,00179834722	1	10	-14,1889591
7	life	764	-0,00174156961	1	4	-5,32223654
8	films	654	-0,00149161811	1	5	-4,877591
9	people	651	-0,00148479827	1	6	-5,799622
10	scene	588	-0,00134154514	1	5	-3,94414282
11	little	564	-0,001286954	1	6	-4,35505247
12	world	524	-0,00119594671	1	5	-3,13338041
13	movies	499	-0,00113905279	1	6	-3,410324
14	scenes	485	-0,00110718724	1	6	-3,22191477
15	don	469	-0,00107076531	1	3	-1,50656676
16	love	459	-0,00104799948	1	4	-1,9241271
17	plot	438	-0,00100018515	1	4	-1,75232434
18	makes	435	-0,0009933539	1	5	-2,16054463
19	doesn	429	-0,000979691	1	5	-2,10143733
20	audience	417	-0,0009523632	1	8	-3,17708349
21	re	411	-0,0009386983	1	2	-0,77161
22	performance	405	-0,000925032829	1	11	-4,12102127

Figure 1: Results ranked by frequency of the term.

As one can see, a ranking based on the frequency of a term does not provide us with much information except for the fact that the source contains a lot of data about films, people, time and stories. When we relate the top terms to each other we can conclude it is about movies but no extra knowledge is extracted.

Rank	Term	Frequency	Score	Count	Lenght	Bval
114	special effects	175	-0,153191164	2	15	-402,1268
495	pulp fiction	79	-0,153222054	2	12	-145,2545
682	supporting cast	63	-0,153227255	2	15	-144,799759
345	star wars	99	-0,153215662	2	9	-136,515152
736	science fiction	59	-0,153228521	2	15	-135,607239
1	film	3849	-0,008666238	1	4	-133,4254
825	phantom menace	54	-0,153230131	2	14	-115,84198
294	ve seen	107	-0,153213114	2	7	-114,756622
838	motion picture	53	-0,153230444	2	14	-113,696991
1076	starship troopers	43	-0,153233677	2	17	-112,013817
999	romantic comedy	46	-0,1532327	2	15	-105,73056
529	star trek	76	-0,153223038	2	9	-104,804558
881	boogie nights	51	-0,1532311	2	13	-101,592216
554	real life	72	-0,153224453	2	9	-99,289444
859	jackie brown	52	-0,153230771	2	12	-95,6160049
1148	writer director	41	-0,153234348	2	15	-94,23912
1783	character develo...	28	-0,15323849	2	21	-90,10423
1266	main characters	38	-0,153235346	2	15	-87,34415
897	jackie chan	50	-0,153231412	2	11	-84,2772751
1267	subject matter	38	-0,153235272	2	14	-81,5211639
1527	action sequences	32	-0,153237253	2	16	-78,45747
1232	austin powers	39	-0,153234944	2	13	-77,69012

Figure 2: Results ranked by the value of the B Ranking-Method [1]. Note the term "film" appears to often to ignore.

The results of the B Ranking-Method are more promising. It is now clear that the data is about movies when we look at the top seven terms on list and try to relate them to each other it also provides us with a lot more knowledge then our previous results. We can conclude that there is a lot of writing about special effects, science fiction movies and pulp fiction. Relating that back to the top term "film" we can conclude the most popular movies in this stack of documents are science fiction movies like "Star Wars" and "Star Trek", but also other motion pictures like "Pulp fiction" or "Romantic comedy" are popular. Unfortunately, we did not discover if this popularity is in positive or negative context.

[1] The rank column shows the rank based on figure 4 (Term frequency)

Rank	Term	Frequency	Score	Count	Lenght	Bval
2881	/	19	0	1	1	0
5959	]	9	0	1	1	0
6698	\	8	0	1	1	0
9099	di	5	-1,25661436E-05	1	2	-0,000125661434
9187	hy	5	-1,25661436E-05	1	2	-0,000125661434
9267	ee	5	-1,25661436E-05	1	2	-0,000125661434
9348	dj	5	-1,25661436E-05	1	2	-0,000125661434
9500	$5	5	-1,25661436E-05	1	2	-0,000125661434
9580	$1	5	-1,25661436E-05	1	2	-0,000125661434
9621	#1	5	-1,25661436E-05	1	2	-0,000125661434
9933	um	5	-1,25661436E-05	1	2	-0,000125661434
9950	ca	5	-1,25661436E-05	1	2	-0,000125661434
9963	ha	5	-1,25661436E-05	1	2	-0,000125661434
7877	rd	6	-1,485084E-05	1	2	-0,000178210073
7901	iq	6	-1,485084E-05	1	2	-0,000178210073
8034	'n	6	-1,485084E-05	1	2	-0,000178210073
8055	uk	6	-1,485084E-05	1	2	-0,000178210073
8180	hi	6	-1,485084E-05	1	2	-0,000178210073
8525	#2	6	-1,485084E-05	1	2	-0,000178210073
9268	ohh	5	-1,25661436E-05	1	3	-0,000188492151
9395	92t	5	-1,25661436E-05	1	3	-0,000188492151
9417	yup	5	-1,25661436E-05	1	3	-0,000188492151

Figure 3: Terms with Bval 0

As one can see, these terms passed the stop list filter or are terms that not exist. It gives us insight in how to tune the settings of the B Ranking-Method.

As expected, simply counting the frequency of words does not tell us much. The top terms are all single words it is difficult to explain what the documents are about. As one can see in figure 4, the only facts retrieved is that the document collection is about movies. Also some of the single word terms like “don” and “re” does not make any sense at all and we cannot put them in any context.

So can we conclude that this type of text mining algorithm is not useful? We disagree; the text mining algorithm has produced something interesting. Take a look at the other data in the columns count and score. A first look at those columns does not tell us anything, however this information can be used for further computing.

When we take a look at the terms that are ranked based on the computed Bval:

$$Bval(t) = Tl(t) * Tf(t) * Tw(t)$$

where $Tl(t)$ denotes the Length column, $Tf(t)$ denotes the frequency column and $Tw(t)$ denotes the weight which is given by the text mining algorithm. Since the used algorithm gives us a high negative score for a relevant term which appears a lot, the most relevant terms based on the Bval are the terms with the highest negative score (lowest scores). An important note on this is that the weight of a term must be relevant to its occurrence within the corpus compared to other terms found.
According to the B Ranking-Method the most relevant term is “Special effects”, which puts the term at position 1 (Figure 2: Results ranked by the value of the B Ranking-Method) . Based on frequency this term is ranked 114 in figure 1: Results ranked by frequency of the term.
Figure 3: Terms with Bval 0, shows us the terms with a weight of 0. As we can see these are not valid terms and should be included in the stop list.

Based on the 50 reviews we randomly selected and read, we felt in line with the trend given by the B Ranking-Method. Most reviews where about science fiction we found several sequels of science fiction movies and some of them refer to other science fiction movies. The term “ve seen” is wrongly placed in the list because the stop list we set did remove ‘ and changed broke “I’ve seen” into “I ve seen” where “I” is a stop word. For this reason we cannot blame the text mining algorithm to consider “ve seen” as a two word term. We do not found it necessary to change the stop list, add a filter and redo all the steps to compute results again. Instead we chose to ignore the ranking of this term.
The term “Film” appears so many times it cannot be ignored.

6.2 Second Experiment

Not completely unsatisfied with the results from the initial experiment a second experiment was setup to test the B Ranking-Method on different corpuses and varying its size. We added two more corpuses: a data set consisting of 129,000 abstracts describing NSF awards for basic research http://archive.ics.uci.edu/ml/databases/nsfabs/Part1.zip and the titles of every paper to appear in the Proceedings of the National Academy of Sciences (USA) from its inception in 1915 until March 2005 http://www.cs.nyu.edu/~roweis/data/pnas_all.tar (about 80,000 papers) we also decided to re-run the Movie Review corpus again but this time varying its size, from 100 to 1000. We changed the minimal occurrence of a term to five and changed our implementation of the scoring algorithm to a positive log.

6.3 Results Second Experiment

Abstracts “awards_1990\awd_1990_00” documents.

Term	Frequency	Score	Count	Lenght	Bval
expires	2212	0,003332525	1	7	51,6241455
file	2162	0,00325721363	1	4	28,1814117
title	1140	0,00171785068	1	5	9,800338
start date	803	0,006442018	2	10	51,600563
award	792	0,00119368406	1	5	4,73295736
program ref	758	0,006081233	2	11	50,5715332
amendment date	758	0,006081233	2	14	64,36377
prgm manager	668	0,0053596627	2	12	43,0916862
abstract	654	0,0009858249	1	8	5,16572237
estimated	630	0,0009496755	1	9	5,393207
program	610	0,000919550948	1	7	3,9329195
estimated investigator	580	0,004654127	2	23	62,300148
type	574	0,000865326845	1	4	1,99025178
principal investigator current	557	0,007826374	3	30	131,0135
research	548	0,000826164964	1	8	3,62851644
standard grant	424	0,003403406	2	14	20,2979126
principal investigator	412	0,00330719654	2	22	30,1219463
oceanography	404	0,000609268434	1	12	2,96104455
chemistry	394	0,000594206154	1	9	2,112403
students	388	0,0005851688	1	8	1,82104528
current sponsor	381	0,00305865565	2	16	18,7434425
nsf org	379	0,00304262084	2	7	8,029476
award nsf org	379	0,00532754976	3	13	26,0410633
estimated expected total	379	0,00532754976	3	25	50,078968
award nsf	379	0,00304262084	2	9	10,3236122
prgm	379	0,000571612734	1	4	0,864278436
award instr	379	0,00304262084	2	11	12,6177483
estimated expected	379	0,00304262084	2	19	21,7942924
investigator current sponsor	374	0,005257358	3	29	56,86884
applications nec	346	0,00277804513	2	16	15,29036
continuing grant	324	0,00260166125	2	16	13,5702648
mathematics	323	0,0004872641	1	11	1,72588944

Figure 4: Results ranked by frequency of the term.

It is interesting to see that multi word terms appear in the top 10 terms based on the frequency of the term. The term, sub term “estimated“ appears a lot in the corpus. This is because each document has these terms in their headers.

	Term	Frequency	Score	Count	Lenght	Bval
	principal investigator current	557	0,007826374	3	30	131,0135
	start date	803	0,006442018	2	10	51,600563
	program ref	758	0,006081233	2	11	50,5715332
	amendment date	758	0,006081233	2	14	64,36377
	prgm manager	668	0,0053596627	2	12	43,0916862
	award nsf org	379	0,00532754976	3	13	26,0410633
	estimated expected total	379	0,00532754976	3	25	50,078968
	investigator current sponsor	374	0,005257358	3	29	56,86884
	estimated investigator	580	0,004654127	2	23	62,300148
	standard grant	424	0,003403406	2	14	20,2979126
	expires	2212	0,003332525	1	7	51,6241455
	principal investigator	412	0,00330719654	2	22	30,1219463
	file	2162	0,00325721363	1	4	28,1814117
	current sponsor	381	0,00305865565	2	16	18,7434425
	nsf org	379	0,00304262084	2	7	8,029476
	award nsf	379	0,00304262084	2	9	10,3236122
	award instr	379	0,00304262084	2	11	12,6177483
	estimated expected	379	0,00304262084	2	19	21,7942924
	applications nec	346	0,00277804513	2	16	15,29036
	continuing grant	324	0,00260166125	2	16	13,5702648
	co principal investigator	180	0,00253392	3	25	11,5926847
	mps direct	218	0,001751812	2	10	3,78391385
	title	1140	0,00171785068	1	5	9,800338
	edu principal	200	0,00160749792	2	13	4,22128963
	geo directorate	194	0,00155939325	2	15	4,49105263
	physical scien start	109	0,00153719808	3	20	3,3818357
	scien start date	109	0,00153719808	3	16	2,70546865
	investigator current	183	0,00147120131	2	20	5,325749
	co principal	182	0,0014631839	2	12	3,16047716
	nsf program	170	0,00136697455	2	11	2,526169
	ocean sciences	155	0,00124671287	2	14	2,670459
	oce division	152	0,00122266053	2	12	2,25947666

Figure 5: Results ranked by score.

Knowing the headers of the documents in the corpus it is interesting to see how the results are ranked differently by the score. Because the multi word terms appear frequent their respective language models are bigger and they receive a better score thus; pushing single word terms down the list.

	Term	Frequency	Score	Count	Lenght	Bval
	principal investigator current	557	0,007826374	3	30	131,0135
	amendment date	758	0,006081233	2	14	64,36377
	estimated ▯investigator	580	0,004654127	2	23	62,300148
	investigator current ▯sponsor	374	0,005257358	3	29	56,86884
	expires	2212	0,003332525	1	7	51,6241455
	start date	803	0,006442018	2	10	51,600563
	program ref	758	0,006081233	2	11	50,5715332
	estimated ▯expected total	379	0,00532754976	3	25	50,078968
	prgm manager	668	0,0053596627	2	12	43,0916862
	principal investigator	412	0,00330719654	2	22	30,1219463
	file	2162	0,00325721363	1	4	28,1814117
	award nsf org	379	0,00532754976	3	13	26,0410633
	estimated ▯expected	379	0,00304262084	2	19	21,7942924
	standard grant	424	0,003403406	2	14	20,2979126
	current ▯sponsor	381	0,00305865565	2	16	18,7434425
	applications nec	346	0,00277804513	2	16	15,29036
	continuing grant	324	0,00260166125	2	16	13,5702648
	award instr	379	0,00304262084	2	11	12,6177483
	co principal investigator	180	0,00253392	3	25	11,5926847
	award nsf	379	0,00304262084	2	9	10,3236122
	title	1140	0,00171785068	1	5	9,800338
	nsf org	379	0,00304262084	2	7	8,029476
	estimated	630	0,0009496755	1	9	5,393207
	investigator current	183	0,00147120131	2	20	5,325749
	abstract	654	0,0009858249	1	8	5,16572237
	award	792	0,00119368406	1	5	4,73295736
	geo directorate	194	0,00155939325	2	15	4,49105263
	edu principal	200	0,00160749792	2	13	4,22128963
	program	610	0,000919550948	1	7	3,9329195
	mps direct	218	0,001751812	2	10	3,78391385
	research	548	0,000826164964	1	8	3,62851644
	mathematical sciences	147	0,00118257327	2	21	3,6009357

Results ranked by the value of the B Ranking-Method.

Compared to the results of the score we expected a further refinement of the data by pushing context less single word terms even further down the list, however; this was not entirely the case. On one hand it improved the position of certain multi term words but also ranked some single word terms higher on the list.

All Titles of the National Academy of Sciences (USA) corpus.

Term	Frequency	Score	Count	Lenght	Bval
nbsp	1516	0,00124255661	1	4	7,53983355
escherichia coli	1390	0,001032565	2	16	22,9312038
protein	1067	0,0008746648	1	7	6,52674866
role	988	0,000809935562	1	4	3,20410514
gene	938	0,0007689676	1	4	2,88824224
evidence	881	0,000722264231	1	8	5,08474
human	879	0,0007206255	1	5	3,163546
expression	858	0,000703419	1	10	6,042369
cells	775	0,000635412231	1	5	2,45904541
proteins	770	0,000631315459	1	8	3,8939538
identification	726	0,0005952637	1	14	6,05859375
structure	719	0,0005895282	1	9	3,80953121
dna	705	0,0005780572	1	3	1,22085679
mechanism	675	0,00055347645	1	9	3,35738826
activity	643	0,000527257	1	8	2,70799184
effects	640	0,0005247989	1	7	2,35477257
induced	629	0,000515785941	1	7	2,267395
activation	623	0,000510869839	1	10	3,1776104
effect	622	0,00051005045	1	6	1,90656853
regulation	621	0,0005092311	1	10	3,157233
cover	615	0,000504314958	1	5	1,54824686
function	599	0,0004912052	1	8	2,34992576
vivo	594	0,000487108424	1	4	1,15931809
genes	586	0,000480553572	1	5	1,41042471
formation	581	0,000476456771	1	9	2,48710442
binding	576	0,00047235997	1	7	1,907862
human immunodeficiency virus	596	0,0004618757	3	28	7,746579
interaction	563	0,000461708318	1	11	2,85428071
associated	554	0,0004543341	1	10	2,52155423
specific	553	0,000453514745	1	8	2,002721
inhibition	534	0,000437946932	1	10	2,34301615
model	531	0,000435488852	1	5	1,15404546

Results ranked by score.

When scoring the titles corpus the ranking of terms based on the score there is only one multi word term in the 10 top terms list. The list provides us with some context from the single word terms but this is mainly because the terms are domain specific.

Term	Frequency	Score	Count	Lenght	Bval
escherichia coli	1390	0,001032565	2	16	22,9312038
human immunodeficiency virus	596	0,0004618757	3	28	7,746579
nbsp	1516	0,00124255661	1	4	7,53983355
protein	1067	0,0008746648	1	7	6,52674866
identification	726	0,0005952637	1	14	6,05859375
expression	858	0,000703419	1	10	6,042369
evidence	881	0,000722264231	1	8	5,08474
proteins	770	0,000631315459	1	8	3,8939538
structure	719	0,0005895282	1	9	3,80953121
characterization	517	0,000424017839	1	16	3,50069118
mechanism	675	0,00055347645	1	9	3,35738826
major histocompatibility complex	363	0,000281461544	3	32	3,242437
role	988	0,000809935562	1	4	3,20410514
activation	623	0,000510869839	1	10	3,1776104
human	879	0,0007206255	1	5	3,163546
regulation	621	0,0005092311	1	10	3,157233
gene	938	0,0007689676	1	4	2,88824224
interaction	563	0,000461708318	1	11	2,85428071
activity	643	0,000527257	1	8	2,70799184
associated	554	0,0004543341	1	10	2,52155423
gene expression	473	0,000351614173	2	15	2,505251
formation	581	0,000476456771	1	9	2,48710442
cells	775	0,000635412231	1	5	2,45904541
development	514	0,000421559758	1	11	2,38813615
effects	640	0,0005247989	1	7	2,35477257
function	599	0,0004912052	1	8	2,34992576
inhibition	534	0,000437946932	1	10	2,34301615
induced	629	0,000515785941	1	7	2,267395
drosophila melanogaster	358	0,00026621687	2	23	2,17978382
saccharomyces cerevisiae	350	0,000260276167	2	24	2,17382646
specific	553	0,000453514745	1	8	2,002721
binding	576	0,00047235997	1	7	1,907862

Results ranked by the value of the B Ranking-Method.

When we rank the terms on the B Ranking-Method more context about the source documents is provided, however; there are still a lot of single term words in the top 10 ranking and we discovered this makes it hard to convert the results into knowledge about the source content.

Movie Review (1000)

	Term	Frequency	Score	Count	Lenght	Bval
	film	3849	0,00381977065	1	4	58,79391
	movie	1950	0,00193543651	1	5	18,8801823
	time	948	0,000941174862	1	4	3,57269979
	story	869	0,0008627849	1	5	3,74448657
	character	821	0,0008151556	1	9	6,01584864
	characters	789	0,000783402764	1	10	6,173214
	life	764	0,0007585958	1	4	2,32130313
	films	654	0,000649445341	1	5	2,12693357
	people	651	0,000646468543	1	6	2,52122736
	scene	588	0,0005839551	1	5	1,71974778
	little	564	0,0005601404	1	6	1,898876
	world	524	0,000520449365	1	5	1,36617959
	movies	499	0,0004956424	1	6	1,48097956
	scenes	485	0,000481750525	1	6	1,39900351
	don	469	0,0004658741	1	3	0,654087245
	love	459	0,000455951318	1	4	0,8353028
	plot	438	0,0004351135	1	4	0,7640593
	makes	435	0,000432136672	1	5	0,9377366
	doesn	429	0,000426183018	1	5	0,912031651
	audience	417	0,000414275681	1	8	1,37870944
	re	411	0,000408322026	1	2	0,334824055
	performance	405	0,000402368372	1	11	1,788125
	role	391	0,000388476474	1	4	0,6060233
	actually	376	0,000373592338	1	8	1,12675452
	played	359	0,0003567236	1	6	0,7662423
	director	346	0,000343824	1	8	0,954455435
	look	345	0,000342831743	1	4	0,4717365
	family	339	0,0003368781	1	6	0,683188736
	comes	335	0,000332908967	1	5	0,555958
	takes	328	0,000325963018	1	5	0,536209166
	isn	326	0,0003239785	1	3	0,3178229
	plays	325	0,0003229862	1	5	0,523237646

Results ranked by score.

Again, the top terms in the list are single word terms. The corpus reveals no clue about its contents apart from the, already known fact, that its contents is mainly about films and movies.

Term	Frequency	Score	Count	Lenght	Bval
film	3849	0,00381977065	1	4	58,79391
movie	1950	0,00193543651	1	5	18,8801823
characters	789	0,000783402764	1	10	6,173214
character	821	0,0008151556	1	9	6,01584864
story	869	0,0008627849	1	5	3,74448657
time	948	0,000941174862	1	4	3,57269979
people	651	0,000646468543	1	6	2,52122736
life	764	0,0007585958	1	4	2,32130313
films	654	0,000649445341	1	5	2,12693357
little	564	0,0005601404	1	6	1,898876
performance	405	0,000402368372	1	11	1,788125
scene	588	0,0005839551	1	5	1,71974778
movies	499	0,0004956424	1	6	1,48097956
scenes	485	0,000481750525	1	6	1,39900351
audience	417	0,000414275681	1	8	1,37870944
world	524	0,000520449365	1	5	1,36617959
actually	376	0,000373592338	1	8	1,12675452
director	346	0,000343824	1	8	0,954455435
makes	435	0,000432136672	1	5	0,9377366
doesn	429	0,000426183018	1	5	0,912031651
love	459	0,000455951318	1	4	0,8353028
played	359	0,0003567236	1	6	0,7662423
plot	438	0,0004351135	1	4	0,7640593
original	302	0,0003001638	1	8	0,727597058
especially	262	0,000260472734	1	10	0,6850433
family	339	0,0003368781	1	6	0,683188736
don	469	0,0004658741	1	3	0,654087245
performances	233	0,0002316967	1	12	0,6450436
role	391	0,000388476474	1	4	0,6060233
course	316	0,0003140557	1	6	0,597333968
relationship	216	0,000214827989	1	12	0,559412062
comes	335	0,000332908967	1	5	0,555958

Results ranked by the value of the B Ranking-Method.

When the results are ranked by the B Ranking-Method. There is a shift in the ranking but the multi word terms do not appear in the top term list.

Movie Review (100)

	Term	Frequency	Score	Count	Lenght	Bval
	film	424	0,00419704849	1	4	7,13498259
	movie	205	0,00203178683	1	5	2,0724225
	time	110	0,001092518	1	4	0,485077977
	character	106	0,00105296983	1	9	1,01401
	story	92	0,000914551259	1	5	0,425266325
	life	90	0,0008947772	1	4	0,325698882
	characters	88	0,0008750031	1	10	0,778752744
	films	86	0,000855229	1	5	0,3720246
	plot	64	0,0006377141	1	4	0,165805668
	movies	63	0,0006278271	1	6	0,233551666
	little	63	0,0006278271	1	6	0,233551666
	people	62	0,000617940037	1	6	0,233581334
	world	60	0,000598165963	1	5	0,182440624
	makes	57	0,000568504853	1	5	0,159181356
	performance	56	0,0005586178	1	11	0,350253373
	don	55	0,0005487308	1	3	0,08889439
	scenes	55	0,0005487308	1	6	0,177788779
	doesn	53	0,000528956647	1	5	0,137528732
	actually	51	0,0005091826	1	8	0,203673035
	role	50	0,000499295536	1	4	0,101856291
	scene	50	0,000499295536	1	5	0,127320364
	re	48	0,000479521463	1	2	0,0469931029
	plays	48	0,000479521463	1	5	0,117482759
	john	43	0,000430086278	1	4	0,0722544938
	love	43	0,000430086278	1	4	0,0722544938
	director	42	0,0004201992	1	8	0,144548535
	comes	42	0,0004201992	1	5	0,09034283
	action	41	0,000410312176	1	6	0,09847492
	audience	41	0,000410312176	1	8	0,1312999
	actor	40	0,000400425139	1	5	0,08208715
	family	40	0,000400425139	1	6	0,09850458
	job	39	0,0003905381	1	3	0,044521343

Results ranked by score.

When we reduce the size of the corpus, there is little change in the results when we rank the terms by their scores.

Term	Frequency	Score	Count	Lenght	Bval
film	424	0,00419704849	1	4	7,13498259
movie	205	0,00203178683	1	5	2,0724225
character	106	0,00105296983	1	9	1,01401
characters	88	0,0008750031	1	10	0,778752744
time	110	0,001092518	1	4	0,485077977
story	92	0,000914551259	1	5	0,425266325
films	86	0,000855229	1	5	0,3720246
performance	56	0,0005586178	1	11	0,350253373
life	90	0,0008947772	1	4	0,325698882
people	62	0,000617940037	1	6	0,233581334
movies	63	0,0006278271	1	6	0,233551666
little	63	0,0006278271	1	6	0,233551666
actually	51	0,0005091826	1	8	0,203673035
world	60	0,000598165963	1	5	0,182440624
scenes	55	0,0005487308	1	6	0,177788779
plot	64	0,0006377141	1	4	0,165805668
makes	57	0,000568504853	1	5	0,159181356
director	42	0,0004201992	1	8	0,144548535
doesn	53	0,000528956647	1	5	0,137528732
audience	41	0,000410312176	1	8	0,1312999
scene	50	0,000499295536	1	5	0,127320364
plays	48	0,000479521463	1	5	0,117482759
role	50	0,000499295536	1	4	0,101856291
family	40	0,000400425139	1	6	0,09850458
action	41	0,000410312176	1	6	0,09847492
performances	29	0,0002916677	1	12	0,09800035
original	35	0,000350989954	1	8	0,0954692662
comes	42	0,0004201992	1	5	0,09034283
played	39	0,0003905381	1	6	0,0890426859
don	55	0,0005487308	1	3	0,08889439
hollywood	31	0,000311441778	1	9	0,08408928
instead	34	0,0003411029	1	7	0,0835702047

Results ranked by the value of the B Ranking-Method

The results ranked by the B Ranking-Method show little change when the corpus size is reduced.

6.4 Third experiment.

When studying the results of our second experiment we discovered a valuable hint about context terms and non-context terms. When giving thought to our research objective we came up with a new idea based on the following relation: The relevance of terms should be based on the context it has. It became apparent to us that single word terms carry no context at all. Single word terms like “movie” or “award” does not provide us with any knowledge, however; Multi-word terms like “action movie” or “nsf award” does.

Based on our thoughts from the second experiment we decided to run another experiment where we de-coupled the B Ranking-Method scoring algorithm from the scoring mechanism and put the weight of the score of a term to its length. We also decided to exclude single word terms. The algorithm was changed into the following:

$$Bval = \log(Wl(t) * Tl(t)) * Tf(t) \;\; AND \; Tl(t) > 1$$

where $Wl(t)$ denotes the number of characters in a term t, $Tl(t)$ the number of words in a term t and $Tf(t)$ the frequency of a term t in the corpus. with a minimal term occurrence of five.

In order to make a proper comparison we decided to run the algorithm twice, once including single term-words and once more to include them. We then ran the algorithm on the Movie Review (1000) corpus and it produced the following results:

	Term	Frequency	Score	Count	Lenght	Bval
	film	4248	0,00421568938	1	4	9,740498
	movie	2151	0,00213488424	1	5	9,283126
	time	1056	0,00104834081	1	4	8,348537
	story	964	0,0009570513	1	5	8,480529
	character	887	0,000880645937	1	9	8,985069
	characters	856	0,000849885342	1	10	9,054855
	life	829	0,000823093869	1	4	8,106515
	films	714	0,000708982	1	5	8,180321
	people	693	0,0006881442	1	6	8,332789
	scene	668	0,00066333724	1	5	8,113726
	little	631	0,000626623	1	6	8,239065
	world	559	0,000555179	1	5	7,93558741
	movies	551	0,000547240837	1	6	8,103495
	scenes	531	0,0005273953	1	6	8,066522
	doesn	511	0,000507549732	1	5	7,84580755
	performance	492	0,0004886965	1	11	8,596374
	don	491	0,0004877042	1	3	7,29505634
	plot	489	0,000485719647	1	4	7,57865667
	love	478	0,0004748046	1	4	7,555905
	makes	476	0,000472820044	1	5	7,77485561
	re	461	0,000457935879	1	2	6,82654524
	audience	439	0,0004361058	1	8	8,163941
	role	439	0,0004361058	1	4	7,47079372
	director	427	0,000424198457	1	8	8,136226
	actually	401	0,00039839925	1	8	8,073403
	played	392	0,000389468769	1	6	7,76302147
	family	376	0,000373592338	1	6	7,721349
	plays	367	0,000364661828	1	5	7,51479959
	look	365	0,0003626773	1	4	7,286192
	comes	356	0,0003537468	1	5	7,484369
	takes	351	0,0003487854	1	5	7,470224
	funny	349	0,000346800836	1	5	7,46451

Results ranked by score.

As before the ranking based on the scoring method provides us little knowledge of the content of the corpus.

Term	Frequency	Score	Count	Lenght	Bval
film	4248	0,00421568938	1	4	5888,97852
movie	2151	0,00213488424	1	5	3461,901
characters	856	0,000849885342	1	10	1971,01282
character	887	0,000880645937	1	9	1948,93823
story	964	0,0009570513	1	5	1551,49817
time	1056	0,00104834081	1	4	1463,92688
people	693	0,0006881442	1	6	1241,68933
performance	492	0,0004886965	1	11	1179,76453
life	829	0,000823093869	1	4	1149,238
films	714	0,000708982	1	5	1149,13867
little	631	0,000626623	1	6	1130,60022
scene	668	0,00066333724	1	5	1075,10449
movies	551	0,000547240837	1	6	987,25946
scenes	531	0,0005273953	1	6	951,424255
audience	439	0,0004361058	1	8	912,8748
world	559	0,000555179	1	5	899,6758
director	427	0,000424198457	1	8	887,9215
actually	401	0,00039839925	1	8	833,8561
doesn	511	0,000507549732	1	5	822,4228
makes	476	0,000472820044	1	5	766,092468
played	392	0,000389468769	1	6	702,3697
original	330	0,000327947579	1	8	686,2157
plot	489	0,000485719647	1	4	677,897949
performances	272	0,0002703955	1	12	675,8946
family	376	0,000373592338	1	6	673,701538
love	478	0,0004748046	1	4	662,6487
especially	287	0,000285279675	1	10	660,8419
special effects	181	0,000189707513	2	15	615,6167
role	439	0,0004361058	1	4	608,583252
hollywood	272	0,0002703955	1	9	597,6451
plays	367	0,000364661828	1	5	590,6637
comes	356	0,0003537468	1	5	572,9599

Bval with Tl(t)>0

Running the B Ranking-method algorithm against the corpus including single word terms does not provide us with more insight about the corpus context then the selected scoring mechanism.

	Term	Frequency	Score	Count	Lenght	Bval
	saving private ryan	37	5,979283E-05	3	19	19,66607
	robert de niro	26	4,22536E-05	3	14	17,6914616
	tommy lee jones	22	3,58757E-05	3	15	17,3972778
	science fiction films	14	2,31198937E-05	3	21	17,05074
	blair witch project	15	2,47143689E-05	3	19	16,957468
	thin red line	21	3,42812236E-05	3	13	16,8284149
	obi wan kenobi	19	3,10922733E-05	3	14	16,7504883
	natural born killers	12	1,99309434E-05	3	20	16,4419174
	science fiction film	11	1,83364682E-05	3	20	16,1808834
	world war ii	18	2,94977963E-05	3	12	16,1258354
	special effects	181	0,000189707513	2	15	15,8130941
	international film festival	7	1,19585657E-05	3	27	15,7252407
	saturday night live	9	1,51475169E-05	3	19	15,4249907
	drunken master ii	10	1,6741993E-05	3	17	15,4073954
	john cameron mitchell	8	1,35530418E-05	3	21	15,371892
	disney animated feature	7	1,19585657E-05	3	23	15,2442131
	rocky horror picture	8	1,35530418E-05	3	20	15,2255211
	dusk till dawn	11	1,83364682E-05	3	14	15,110858
	waking ned devine	9	1,51475169E-05	3	17	15,0913134
	haley joel osment	9	1,51475169E-05	3	17	15,0913134
	granger movie gauge	8	1,35530418E-05	3	19	15,0716419
	screenwriter kevin williamson	5	8,769615E-06	3	29	14,9302015
	driving miss daisy	8	1,35530418E-05	3	18	14,90944
	original star wars	8	1,35530418E-05	3	18	14,90944
	billy bob thornton	8	1,35530418E-05	3	18	14,90944
	disney animated features	6	1,03640905E-05	3	24	14,90944
	meet joe black	10	1,6741993E-05	3	14	14,8249273
	kenobi ewan mcgregor	7	1,19585657E-05	3	20	14,8249273
	silent bob strike	8	1,35530418E-05	3	17	14,7379646
	limited screen time	7	1,19585657E-05	3	19	14,6710472
	qui gon jinn	11	1,83364682E-05	3	12	14,648406
	gus van sant	10	1,6741993E-05	3	12	14,3624754

Bval with Tl(t)>1

When excluding single word terms and applying our new algorithm for the B ranking-method we can reveal knowledge about the content of the corpus. As you can see when you look at the frequency column, even though some terms appear more frequent then terms ranked higher by the B Ranking-Method. The terms also have various rankings based on the scoring column. When looking to the top terms provided by the B Ranking-Method knowledge about the content of the corpus is revealed. When compared with the other term lists from our third experiment we can say the term list created with the B Ranking-Method where we exclude the single word terms provides us more knowledge then the other lists.

7. Discussion

We start off with a general conclusion. We consider this research to be successful, however; we cannot conclude yet if the B Ranking-Method adds to this success directly. The reason is that more research must be conducted to provide us proof that the B Ranking-Method provides us the information or the fact that excluding single word terms provides us more insight. The relation between information, the number of words in a term and context is useful. Also the redefinition of terms, multi-word terms is one step towards our goal to gain information and acquire insights about the content of large document collections without having to read them. We also defined the following sub questions:

1. Can we extract multi-word terms out of small text document collections?
2. Can we make sense out of multi-word terms without using dictionaries?
3. Can current methods be applied to large collections of small text documents?

We have come to the following conclusions:

Can we extract multi-word terms out of small text document collections?
Using a language model approach to keyphrase extraction (Tomokiyo & Hurst, 2003).

Can we make sense out of multi-word terms without using dictionaries?
By weighing multi-word terms on term count, length and removing single word terms.

Can current methods be applied to large collections of small text documents?

Yes they can. During this research we mined data which contains over one million terms.

We plan to continue our research concerning the B Ranking-Method in the future. We have the feeling that when extracting knowledge from information the focus must lay more on context and less on terms. Maybe single term words can be useful at all? Maybe we should ignore the length of terms and focus purely on the terms or vice versa? Maybe we can optimize our preprocessing and it will result in a much better ranking?

Acknowledgements

I would like to thank TNO for helping me making this research possible.

References

Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics - Special issue on using large corpora*.

Fayyad et al. (1996). *Advances In Knowledge Discovery And Data Mining.* MIT Press Ltd.

Frantzi et al. (1998). Automatic Recognition of Multi-Word Terms the C-value/NC-value method.

Nobata et al. (1999). Automatic Term Identification and Classification in Biology Texts.

Pakal et al. (2002). A Multi-level Text Mining Method to Extract Biological Relationships.

Sager, J. C., Dungworth, D., & McDonald, P. F. (1980). English Special Languages: principles and practice in science and technology. *Oscar Brandstetter Verlag KG*.

SanJuan et al. (2006). Text mining without document context. *Information Processing and Management*, 20.

Tomokiyo, T., & Hurst, M. (2003). A Language Model Approach to Keyphrase Extraction.

Yakushiji, A., Tateisi, Y., Miyao, Y., & Tsujii, J. (2000). Use of a Full Parser for Information Extraction in Molecular Biology Domain. *Genome Informatics 11*, 446–447.

Appendix A

Binomial log likelihood algorithm (Dunning, 1993).

$$-2\log\lambda = 2\left[\log L(p_1,k_1,n_1) + \log L(p_2,k_2,n_2) - \log L(p,k_1,n_1) - \log L(p,k_2,n_2)\right]$$

where

$$\log L(p,n,k) = k\log p + (n-k)\log(1-p)$$

also, $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$ and $p = \frac{k_1+k_2}{n_1+n_2}$.

For the multinomial case, this formulae becomes:

$$-2\log\lambda = 2\left[\log L(P_1,K_1) + \log L(P_2,K_2) - \log L(Q,K_1) - \log L(Q,K_2)\right]$$

where

$$\begin{aligned} p_{ji} &= \frac{k_{ji}}{\sum_j k_{ji}} \\ q_j &= \frac{\sum_i k_{ji}}{\sum_{ij} k_{ji}} \\ \log L(P,K) &= \sum_j k_j \log p_j \end{aligned}$$