Understanding and customizing stopword lists for enhanced patent mapping

Antoine Blanchard

Understanding and customizing stopword lists for enhanced patent mapping*

Antoine Blanchard

Syngenta Crop Protection AG, Intellectual Property Department, Schwarzwaldallee 215, CH-4002 Basel, Switzerland

Abstract

While the use of patent mapping tools is growing, the 'black-box' systems involved do not generally allow the user to interfere further than the preliminary retrieval of documents. Except for one thing: the stopword list, i.e. the list of 'noise' words to be ignored, which can be modified to one's liking and dramatically impacts the final output and analysis. This paper invokes information science and computer science to provide clues for a better understanding of the stopword lists' origin and purpose, and how they fit in the mapping algorithm. Further, it stresses the need for stopword lists that depend on the document corpus analyzed. Thus, the analyst is invited to add and remove stopwords — or even, in order to avoid inherent biases, to use algorithms that can automatically create ad hoc stopword lists.

Key words: text mining, word distribution, Zipf's law, STN AnaVist, Thomson Aureka, OmniViz

1 Introduction

Since the beginning of years 2000's, there has been a remarkable spreading and growth of patent mapping tools, reaching all areas of information searching, information intelligence, technological watch and so on [1,2,3]. A breakthrough can be traced back to 2003 [4, p. 302] when the PIUG Annual Conference in Chicago put a considerable emphasis on analysis and visualization tools and their use in patent information. Patent maps are indeed a category of analysis and visualization tools, also called 'dimensional analysis techniques' [4, p. 174]. Their purpose is to

* Some parts of the material in this paper were first presented at the PIUG Annual Conference 2006 and are used here with permission of PIUG, Inc.

Email address: antoine.blanchard@syngenta.com (Antoine Blanchard).

build two- or three-dimensional representations of data of very high 'dimensionality' (one dimension can be one word, one co-occurrence, one inventor...). Thus, they are useful to visualize complex relationships between items (e.g. documents) and find major patterns while helping to navigate the landscape. They can even become crucial as the amount of patent literature published grows exponentially and managers have to make quick and reliable decisions.

In the context of this article, we are interested in 'concept mapping' tools and methods, based on the so-called 'bag of words' approach that takes words as elementary information units and ignores their meaning or semantic relations. Examples of such commercially available tools are Thomson Aureka®, STN® AnaVist™ or OmniViz®. More precisely, this article focuses on the fundamental step of these tools, namely the lexical analysis that preludes to the mapping of documents according to their content. That is precisely where the stopword list fits in, a small feature having wide repercussions and a critical role in the quality of the output. A glimpse of this importance can be caught when looking at Aureka, whose stopword list contains 1 290 words whereas its subjacent full-text database, Micropatent, has none.

A first aim of the article is to provide the reader with a good understanding of how mapping tools work, with an emphasis on stopwords. Thus, following some historical thoughts (§ 2), the fundamentals of stopword lists in patent mapping are described, resorting to the notions of word distribution (§ 3.1) and scoring algorithm (§ 3.2). A second aim of the article is to give clues for practical applications: preset stopword lists are introduced as the default behaviour (§ 3.3) while the power of user customization is thoroughly discussed (§ 3.4). Beyond these two common options, a last aim of the article is to open perspectives with automated stopword list construction. On this emerging issue, advances from OmniViz and AnaVist are discussed (§ 4.1) in parallel with potential solutions as published in the literature (§ 4.2). We are confident that with this unique mix of theoretical and applied insights, the reader will eventually be able to perform enhanced patent mapping, as the title promises.

2 Stopword lists, then and now

Stopword lists (or 'stoplists') are lists of non information-bearing words as exemplified in Table 1.

They were first introduced in 1958 by Hans Peter Luhn, a computer scientist and information expert who paved the way for automatic indexing and information retrieval. At the International Conference on Scientific Information in Washington DC, he disclosed the Keyword-in-Context (KWIC) indexing technique where keywords are flanked by the left and right contexts, thus discriminating between key-

a	about	above	across	after
afterwards	again	against	all	almost
alone	along	(...)	such	than
that	the	their	them	themselves
then	thence	there	thereafter	(...)

Table 1
Excerpts of classic van Rijsbergen's stopword list [5].

word terms and non-keyword terms. He coined the term 'stopwords' to name the latter. Since they are often occurring words, excluding them from the indexing allowed to reduce the space and time required by 30-50%, a precious advantage at that time. This innovation was adopted and was state of the art in 1979, when van Rijsbergen published the second edition of his seminal book [5] where he suggests a list of 250 stopwords in English. They have since formed a classic stopword list, used by default or as a baseline in text databases.

However, with time, stopwords have been obliterated in patent databases; growing information technology resources and expectations of the searchers, who want every word to be searchable, made them obsolete. Thus, many databases that were reloaded on their host since the 2000's had their stopword list dropped [1] . On the other hand, the apparition and development of mapping tools has unexpectedly fostered the use of stopwords in another context and raised awareness of the information professionals community on this issue.

3 Stopword lists in patent mapping

3.1 Word significance and distribution in corpus

Patent mapping tools are based on another pioneering idea of Luhn that "the frequency of word occurrence in an article furnishes a useful measurement of word significance" [2] [6, p. 160]. Thus, the mapping algorithm works in a two-step process: it first looks at the terms occurring in the corpus and computes a metrics of their significance; then, it maps the documents according to their content in sig-

[1] On STN, such databases include IFIPAT, WPIX, TULSA, INSPEC, MEDLINE, TOXCENTER, FSTA or NTIS. As mentioned in the introduction, it is also the case of Micropatent since January 2007.

[2] Luhn further adds: "The justification of measuring word significance by use-frequency is based on the fact that a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject. This means of emphasis is taken as an indicator of significance."

nificant terms called 'topics'. The very first part is of interest to us, and requires that we look closer at the frequency of words in a corpus. This has been under investigation since the beginning of information science and bibliometrics, but it is a linguist who made the most notable contribution to it: George K. Zipf's law says that the frequency of a word occurring in a body of text or documents is approximately inverse to the rank of that word by the number of occurrences of the most frequently occurring word,

$$P_n \sim \frac{P_1}{n^\alpha}$$

where P_n is the frequency of the n^{th} ranked word, and α is close to 1. For example, the word *the* is the most frequently occurring word in the Syngenta patent portfolio in agricultural biotechnologies [3], occurring 17 186 times. Then, with $\alpha = 1$, we would expect the second-most frequently occurring word *a* to occur 8 593 times (17 186/2). Similarly, we would expect the third-most frequently occurring word *of* to occur 5 729 times (rounded 17 186/3), the fourth-ranked word *to* to occur 4 297 times (rounded 17 186/4), and so on, resulting in a highly skewed distribution with a very long tail made of low frequency occurring words. Figure 1 shows several graphs on the actual word distribution in this corpus vs. the one predicted by Zipf's law. Literature published since 1949 [7] and collected in [8] confirmed that this empiric law is a good account of the actual observations.

Intuitively, and as hypothesized by Luhn in 1958 [6], mid-frequency words in the distribution have both high significance and high discriminating or "resolving" power [4]. They are the very topics or 'clustering concepts' used for mapping. As shown in Figure 2, this means that words above an upper cut-off and below a lower cut-off must be discarded. For this purpose, a scoring algorithm ranks words and thresholds are set arbitrarily to keep only significant and discriminating words [10, § 3.3.3].

3.2 Scoring algorithm

As part of the scoring algorithm, the most common metrics computed is the *term frequency*inverse document frequency* (*tf*idf*) score (or 'weight') assigned to each term [5]. This score was first introduced in 1973 by Salton and Yang [11] and has

[3] This patent portfolio is composed of more than 400 US or EP patents and patent applications — one per Inpadoc family. The lexical analysis was done on the title, abstract and claims of each document.

[4] This has been later corroborated by theoretical studies ([9] to cite but one) and even more by practical success.

[5] We distinguish between 'words' that are used in the text and 'terms' that form the basis for the information system. Terms are typically single words, conflated to their root forms in 'stemming' process. For instance, the words *antibody* and *antibodies* are conflated to one single term, used in scoring and mapping.

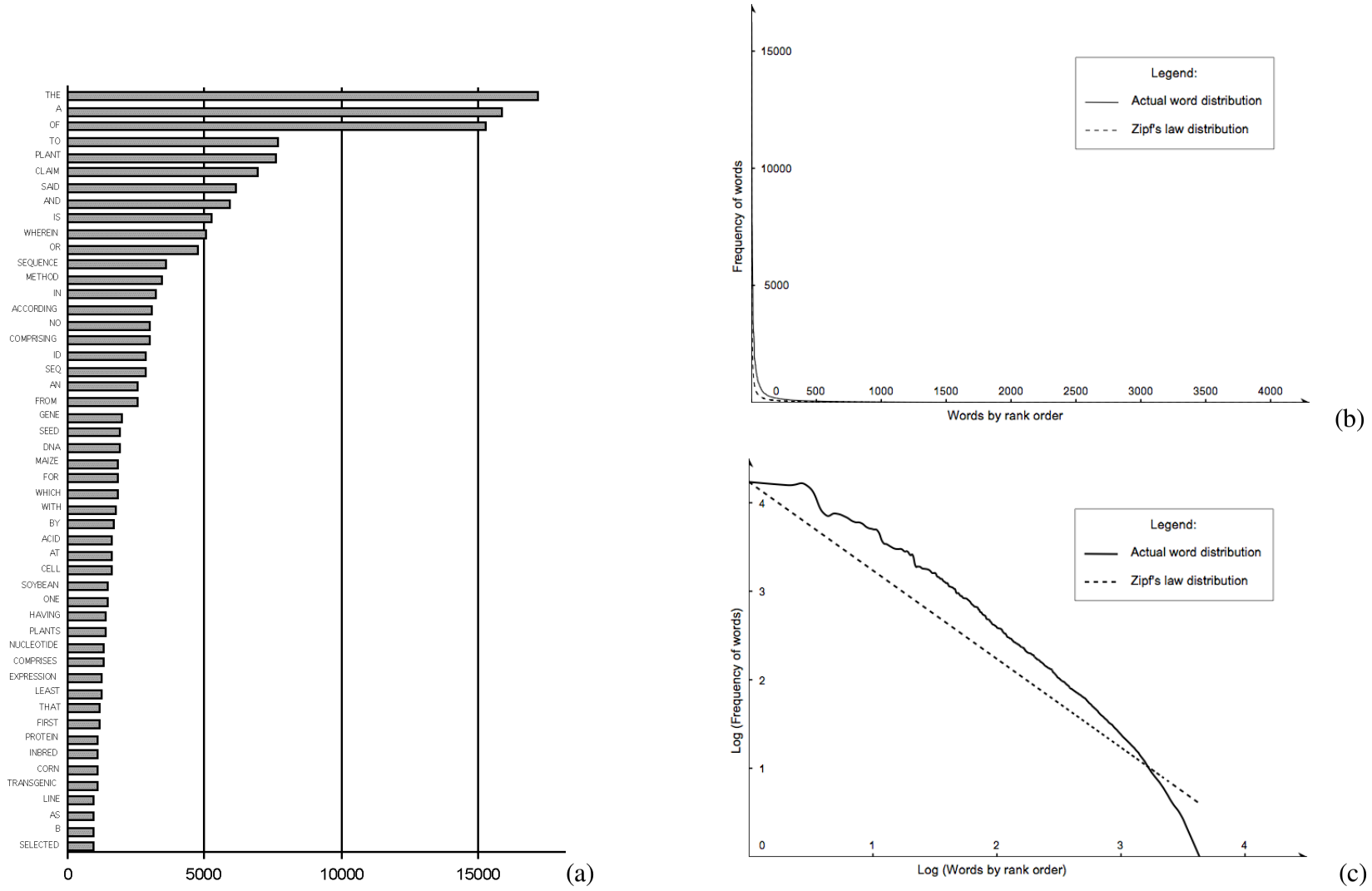


Figure 1. (a) shows the distribution of the most frequently occurring words (top 50 of 6 617 words) in the Syngenta patent portfolio in agricultural biotechnologies. In the corpus, these frequent words account for a large portion of the 298 574 occurrences of words, whereas a large fraction of words appear at a low frequency — including 2 319 'hapax' i.e. words that occur only once. This skewed distribution is well fitted by Zipf's law for $\alpha = 1$ in (b). (For the purpose of this mere example, no attempt was made to adjust α and obtain the best-fitting model. Yet, one should keep in mind that Zipf's law is flexible in this respect.) In (c), both distributions are plotted in a log-log scale that stresses their discrepancy: it is most obvious in the highest and lowest quartiles.

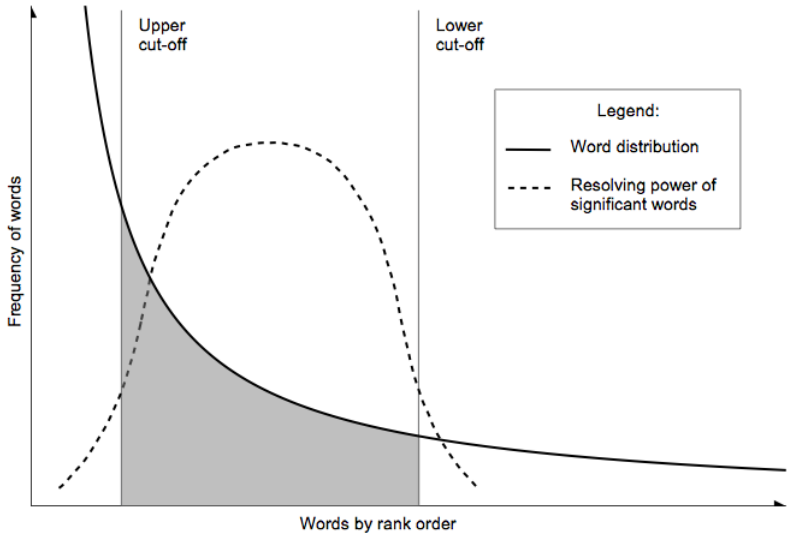


Figure 2. Plot of word distribution showing how topics ideally used in mapping, having both high significance and high discriminating power, fit into Zipf's law. They are comprised between two cut-offs set arbitrarily (in grey). (Republished from [6], with permission.)

received many improvements and adjustments ever since. It is the product of two metrics, term frequency on the one hand and inverse document frequency on the other hand. Term frequency measures the total number of instances of term w in the corpus; it discriminates between frequent and rare words. Inverse document frequency measures the inverse proportion of documents containing the term w in the corpus; it discriminates between function words having a random distribution across documents and content words concentrated in specific documents [12]. Both metrics vary together with the significance of terms and their product was proved to be effective, especially when normalized for document lengths [13]. The terms whose score is beneath a given threshold, either because they occur in too many documents or because they are used very seldom, are considered non discriminant. Therefore, they are removed, leaving the remaining terms as topics [14].

Stopword lists are used beforehand in order to exclude 'noise' words to be ignored by the algorithm [14]. It is a pragmatic and highly useful feature to get smart and significant mapping. However, as it is empiric and subjective, it is also biased and can be misleading. This is why stopword lists need to be understood and used with care.

3.3 Preset vs. user-defined stopword lists

3.3.1 Preset stopword lists

The default behaviour of mapping tools is to handle stopwords as information retrieval systems do and use preset stopword lists, also called 'negative dictionaries' [10, § 3.3.1]. As mentioned in section 2, such dictionaries build on van Rijsbergen's list made of 250 English words. Some have a wide breadth and others are quite limited. For instance, Aureka's stopword list contains 1 290 words covering three major languages (English, French and German) (cf. excerpts in Table 2); on the opposite, OmniViz has a preset list of about 100 stopwords. STN AnaVist has a proprietary stopword list which is hidden to the user. These negative dictionaries include mostly function words that would impede the mapping. However, they face the problem of multi-word phrases — as the common term *in* is ignored, the phrase *oil in water* is no longer taken as a single entity and the map can lose accuracy — and suffer from some irregularities — in Aureka, *his* but not *hers* and *whichever* but not *whatever* are taken as stopwords.

a	aber	abhängig	(. . .)	able
about	above	accordance	according	across
actually	after	again	(. . .)	such
sup	sur	sure	système	t
take	taken	tant	tel	(. . .)

Table 2
Excerpts of Aureka's preset stopword list, to compare with Table 1.

3.3.2 User-defined stopword lists

As a complement to the preset stopword list, all three tools studied in this article (Thomson Aureka, STN AnaVist and OmniViz) allow the user to add words he feels should be in the stopword list since they are "high-frequency function words" [15, p. 71] or "very frequent non-significant words" [16, p. 122]. For instance, patent language (or 'patentese') is not well covered by traditional stopword lists and one may want to add terms such as *embodiment*, *comprising*, *preferred*, *example* etc. Non-specialty terms such as *exhibit*, *develop* or *demonstrate* may be added as well. This is even highly recommended since it can dramatically improve the resulting map [14], all the more as the number of documents increases [17].

As the analyst wants to have the fullest control on his tool, this feature can also be used to add words that are stopwords-like but are specific to the content of a given corpus — which Anthony J. Trippe calls "analysis specific non-content bearing terms" [14, p. 218]. Those words are not traditional function words: they bear information but are trivial when analyzing a corpus on a given topic. Richard K.

Belew calls them ”external keywords”, because they characterize a particular corpus’ content, as opposed to the ”internal keywords” that discriminate documents within the corpus [10, § 3.3.1]. For example, in a corpus of documents dealing with the use of phytase enzyme as feed additive, the word *protein* becomes trivial because enzymes are proteins. Adding this term to the stopword list dramatically changes the map, for a more valuable result (see Figure 3). These kind of words are not exactly stopwords according to the aforementioned definitions. However, they would be encompassed in a broader definition, namely "frequent words that are trivial in a given context". In the remainder of the paper, we will focus on stopwords as in this broader meaning.

3.4 Customizing stopword lists

User-defined stopword lists can be considered as ’customized’ stopword lists. In the coming paragraph, we shall answer different questions pertaining to this customization.

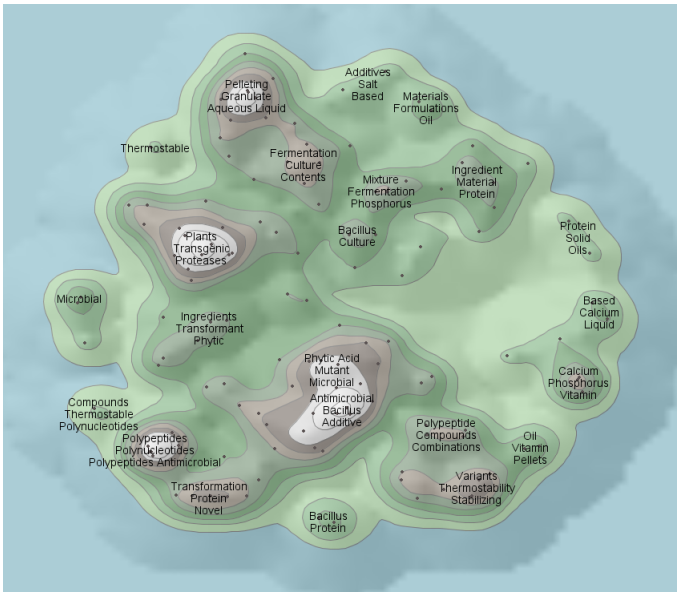
3.4.1 Why?

As demonstrated in the “phytase” example and in [17], customizing stopword lists has a great impact on the mapping output. But why does a single change impact so much on the final map? First, because the consequent change of similarity between documents has some effect and can result in new clusters. Second, because Aureka selects new topics in replacement through dynamic adjustment of cut-offs. Hence a different mapping and labelling, based on less frequent or ’hidden’ words. Third, because heuristics [6] used for clustering, rendering and labelling are sensitive and can have unpredictable and significant effect on the final map. This is particularly true with Aureka.

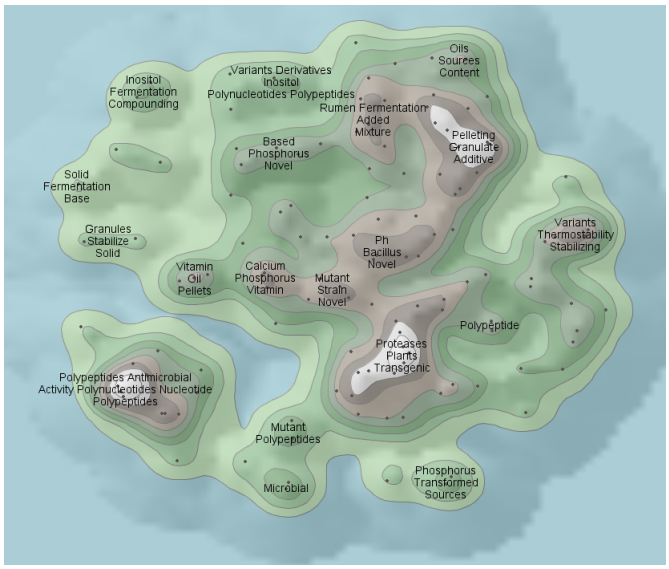
3.4.2 How?

There are hardly ways to customize stopword lists outside trial and error i.e. adding words to the preset stopword list and assessing the result. In Aureka, this empirical procedure can take many rounds as a new stopword list means a new topic list and therefore new trivial words that might be taken as stopwords. At each round, it is recommended to keep the current map, create a separate new map and annotate it with the newest stopwords, in order to keep track of the process [18]. The analyst stops when he feels confident and satisfied with the output. At the end of the process, the stopword list made of the preset and added words should be of

[6] Heuristics are techniques designed to solve problems that ignore whether the solutions can be proven to be correct, but usually produce good solutions.



(a)



(b)

Figure 3. Comparison of two maps obtained in Aureka from a corpus of 132 patent families related to the use of phytase enzyme as feed additive. The word *protein*, present as topic in (a), was added to the stopword list in (b). The latter is less uniform and exhibits new clusters related to *inositol*, *rumen* or *strain*. In both cases, no post-processing e.g. re-labelling was performed.

reasonable size, namely around 400 or 500 words. Authors such as Sinka & Corne [19] emphasized on this optimal size but according to our experience and the one of any Aureka user, ignoring this rule of thumb is not detrimental. Besides, a tool like AnaVist prevents from reckoning that figure by keeping the preset list out of

the analyst's hands.

All three tools allow the analyst to save customized stopword lists for later use. This is useful for often-used stopwords, and is more practical than keeping a list on one's desktop.

3.4.3 When?

The challenge in clustering and mapping is to obtain clusters having low within-group variability and high between-group variability i.e. being homogeneous and well separated [20,21]. Adding trivial i.e. non-discriminating words to the stopword list helps improve the latter criteria. In the "phytase" case, removing *protein* from the topics list enabled discriminating keywords such as *rumen* or *inositol* to show up and helped in reducing the apparent uniformity of the landscape (see Figure 3). Depending on the corpus' topical structure, the need for such adjustment is more or less salient — four of the most often encountered situations are depicted in Figure 4. In this respect and according to our experience, stopword list customization is most needed in the first case i.e. for increasing low between-group variability of maps built around given technological fields.

Information in general and patents in particular have many facets: they can be seen from different perspectives depending on the interest of the analyst. Since mapping produces only one of such perspectives, it is common for the analyst to skew a map that does not suit him in another direction. Stopword list customization is of great help for this purpose. As often exemplified by A. J. Trippe ([22], personal communication), for a map according to pharmaceutical uses rather than chemical names, one would add the latter to the stopword list. This is empirical and imperfect but hardly another feature in mapping tools competes with stopword list customization in this respect.

3.4.4 What quality?

However, when working on this basis, the analyst may lose the objectivity due to the automation. Even if himself gains confidence because he could optimize the map or give it the particular 'flavour' he wanted, another user may feel uncomfortable with the underlying biases — i.e. which stopwords were added and when the optimization process was stopped. Indeed, nothing allows you to control the quality of the output, thus questioning the value of the decisions you can base on it. This could be seen as an advantage since it reminds the end-user that a map is only *one* representation and that hundreds of others could exist. But it is more disturbing to think that by your input, you could hinder knowledge discovery [4, p. 59]. To quote A. J. Trippe, "it is important to let the data speak for itself, as opposed to having the analysis directed by the searcher's preconceived notions" [23]. For overcoming this crucial criticism, one could consider replacing the biased user-input by an

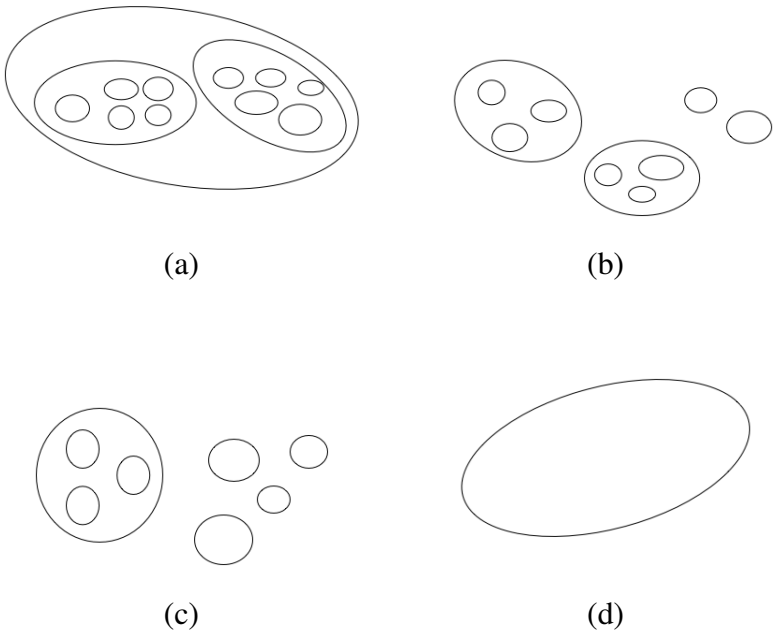


Figure 4. Typical schematic repartitions of map clusters (circles) in four cases. (a): corpus built around a given technological field T results in map showing low within-group variability and low between-group variability; (b): corpus built around a given company C results in map showing average within-group variability and average between-group variability; (c): corpus built around the activity of company C in technological field T results in map showing low within-group variability and high between-group variability; (d): extreme case of corpus built around a given year Y resulting in map showing very high within-group variability and very low between-group variability.

algorithm. This idea has developed for a few years and is more and more to be implemented and used.

4 Toward automated stopword list construction

Automated stopword list construction is undoubtedly the future of stopword lists and has many advantages, including the fact that it can work on any language – what neither the analyst nor the preset stopword list can. We develop further on this approach in the present section, by reviewing tools that have already implemented it and by introducing alternative solutions taken from the literature.

4.1 Existing implementations in mapping tools

Mapping tools developers have been taking on automated stopword list construction after they realized the profits of such renewed approach to stopword lists. In

the first place comes OmniViz that has developed smart concept extraction based on three term sets: Majors Topics, Minor Topics and Other Terms including trivial terms [7] . Proprietary linguistic rules compute the sets on-the-fly based on the actual content of the document corpus. As the Other Terms list complements the default stopword list, one can understand why the latter has no more than a hundred stopwords. All three sets are shown in a single window, called 'Gist table', from which it is then easy to promote or demote terms from one list to the other (see Figure 5). Furthermore, the system supports multi-word phrases so that one can have *a* as Other Term along with *vitamin A* as Major Topic.

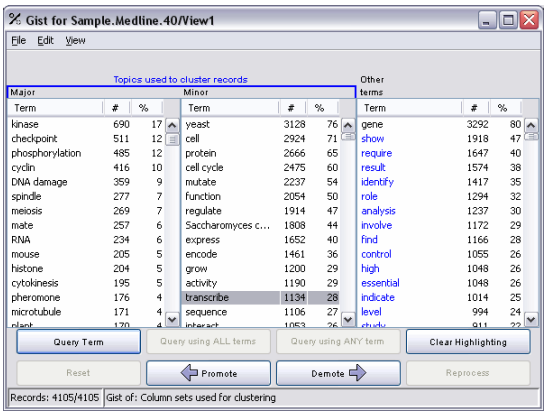


Figure 5. Gist table from OmniViz showing how terms fall into Major Topics, Minor Topics or Other Terms sets.

The 2005-born AnaVist builds on state of the art techniques and also has an advanced algorithm for stopword list construction: in addition to the default stopwords, it computes a list of trivial words to be ignored by the mapping algorithm. In so doing and if the source of the data allows, it can take advantage of the thesaurus and controlled vocabulary from Chemical Abstracts database, which improve accuracy [24]. It is worth noting that, in addition, AnaVist allows the analyst to customize the Concept Frequency (menu Tools > Preferences): when set at 50%, it means that the map is constructed with clustering concepts occurring in no more than 50% of the documents in the document set. This relates to the location of the upper cut-off mentioned in 3.1 and illustrated in Figure 2, with the document frequency taken here as the variable. This is a very powerful feature for customizing concept maps (see example in Figure 6), in line with Luhn's old dream of "alter[ing] the characteristics of the output" by adjusting the threshold location [6, p. 160]. For efficient use, it is recommended "to start high, and then adjust the slider bar downwards" (Brian Sweet, AnaVist Sr. Product Manager, personal communication).

[7] Trivial terms are described by Jeff Saffer as "terms that may be descriptive, but not differentiating" (Jeff Saffer, OmniViz President & CTO, personal communication).

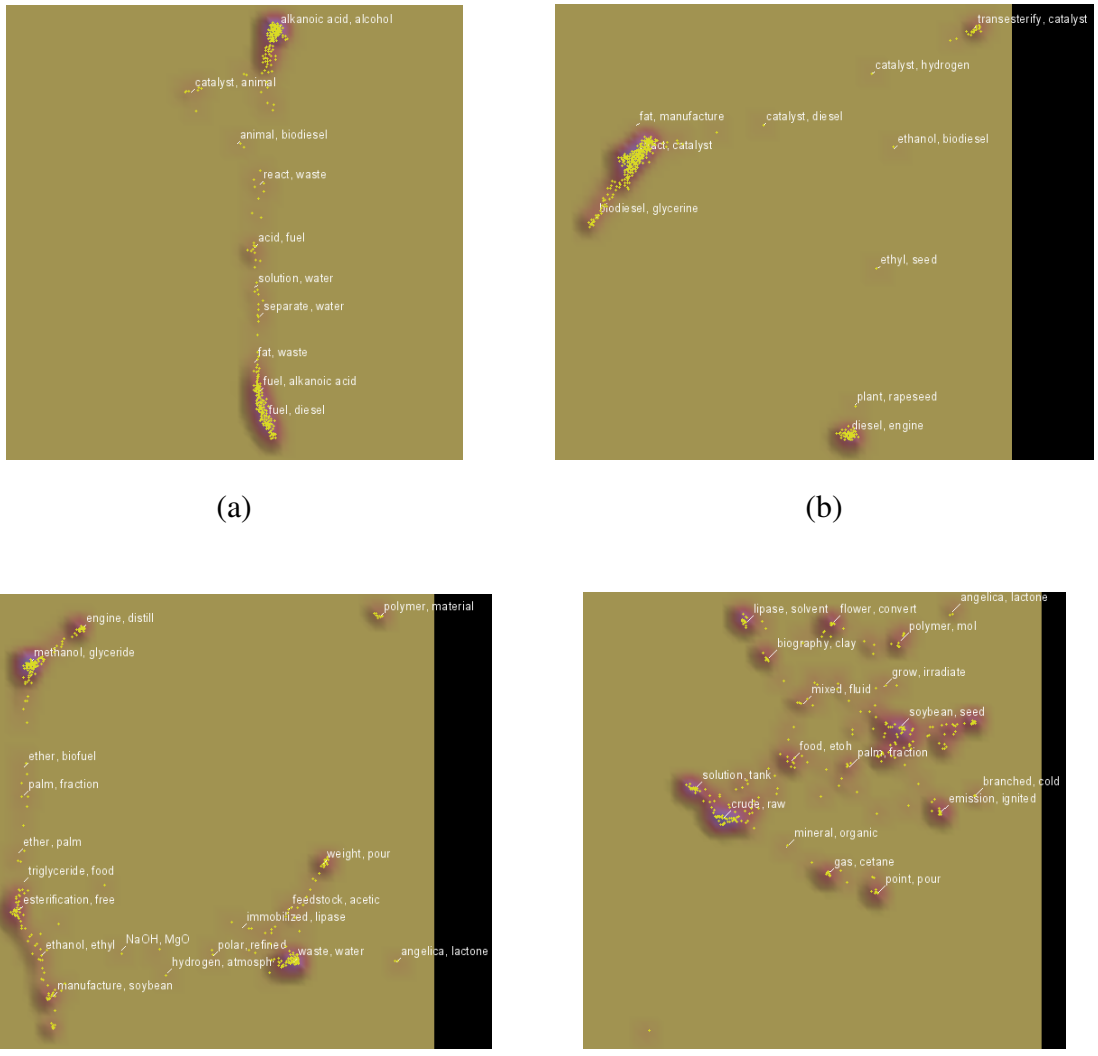


Figure 6. Four instances of AnaVist map on biodiesel obtained when adjusting the Concept Frequency setting. (a): 70% (maximum); (b): 50%; (c): 25% (default); (d): 10% (minimum). From (a) to (d), the clusters dramatically multiply and change from broad (*fuel*, *diesel*, *alcohol*, *catalyst*) to highly specialized (*triglyceride*, *lipase*, *fraction*, *crude*). The dataset composed of 282 records was retrieved from HCAplus by querying patent literature on biodiesel per se, biodiesel mixtures and biodiesel production.

4.2 Solutions published in the literature

The implementations mentioned in the previous paragraph are crucial for achieving enhanced mapping. However, they suffer from opacity and a few limitations. Dedicated to the tool they are embedded in, they make its strengths and weaknesses, with not much freedom left for the analyst. As opposed to these proprietary solutions come algorithms that have been disclosed in the literature. These algorithms could be implemented in companion software capable to work on document corpus

(open-source Terrier [8] for example) and construct stopword lists to be used e.g. in OmniViz; or they could be directly implemented by mapping tools lacking such feature e.g. Aureka. In both cases, their use is subject to experimentation and practice, thus having also a positive effect on analysts trying to acquire experience in patent mapping.

Any scoring algorithm such as *tf*idf* could be considered as relevant for this purpose but it would probably lack refinements and be redundant with the mapping algorithm that it is actually aimed at complementing. Therefore, the examples mentioned thereafter are explicitly designed for the construction of stopword list.

A first algorithm for automated stopword list construction could be the one published by Lo et al. called 'term-based random sampling' [25]. It achieves both good performance and minimal effort by using iterative sampling and computing. A first term is randomly selected in the corpus and all documents containing this term are retrieved. Each term in this document sample is assigned a weight calculated using the Kullback-Leibler divergence measure [26] and normalized by the maximum weight as to be comprised between 0 and 1. All terms are ranked in ascending order of their associated weight and the X least weighted i.e. least informative words are then extracted. After Y iterations, the array of length $X \times Y$ is shrunk by merging the elements containing the same term and associating them with the average of the term's weights. The L least weighted terms are extracted and form the stopword list. X, Y and L are parameters to be determined by trial and error but the authors found the values $X = 200$, $Y = 1\,000$ and $L = 400$ to be optimal. On one of the web corpora tested by the authors, this algorithm picked words such as *environment*, *company*, *industry* or *material* as stopwords. An even better result was achieved when the derived stopword list was merged with a classic stopword list e.g. van Risjbergen's, thus combining common function words and context-specific stopwords.

Other suitable algorithms could be evolutionary algorithms as developed by Sinka and Corne [19]. They consist in 'chromosomes' representing stopword lists built from a pool of 870 candidate stopwords [9], which evolve over 1 000 generations by means of mutation. In each iteration, the performance of the new stopword list is assessed and the corresponding chromosome is kept if it has a better fitness i.e. if it enables more accurate clustering of the documents into the two preset categories "Commercial Banks" and "Soccer". This approach produced better stopword lists that include words such as *technology*, *building*, *reports* or *industry*. Even though this experiment aimed at producing an optimized stopword list once for all, had a control clustering to benchmark against and required high computational efforts, it

[8] Terrier is a software developed at the University of Glasgow, see `http://ir.dcs.gla.ac.uk/terrier/`

[9] The candidate stopwords arise from the union of classic van Rijsbergen's and Brown's stopword lists as well as the top entropy-sorted words from the document corpora analyzed.

might suggest new ways of building ad hoc stopword lists.

5 Conclusion

Mapping tools use word frequency and other statistical metrics fitting Zipf's law to select significant and discriminating words on which they base the visual representation of documents. They are highly useful to gain insight of a technological landscape but can suffer from flaws in the algorithm. As most tools allow the user to modify the preset stopword list, this feature can be extended beyond its intended purpose in order to customize the output. Thus, the analyst can decide to favor one given facet of the document corpus or to reach a certain degree in the clustering detail. Unfortunately, this can introduce biases and somehow jeopardize the meaning and usefulness of the map. That is why we advocate the use of automated approaches that benefit from algorithmic stopword list construction. Some of these approaches can already be found in commercial tools, and thus remain proprietary, while two others mentioned in the paper are taken from the literature: term-based random sampling and evolutionary algorithm. They could be implemented either by skilled analysts or by mapping tools vendors, aiming at enhancing patent mapping. We encourage all of them to tackle this issue and experiment, and eventually share their experience with the whole community.

As the understanding of stopword lists will grow among analysts, other 'technical' issues related to mapping will probably also receive a great deal of attention. To give but one example, the topics selection process could be questioned and confronted with the use of word position and repetition across documents [27]. There is room for improvement almost everywhere, especially when one considers that the methods used (stopword lists, *tf*idf* algorithm) were first developed for information retrieval and *not* analysis. Patent mapping is a field on the move and all people involved, users as well as developers, need to work together to make it more efficient and reliable.

Acknowledgements

I thank Anthony J. Trippe for insightful discussions and comments on earlier draft. I am indebted to Brian Sweet, Jeffrey Saffer and Larry Cady for information on AnaVist, OmniViz and Aureka respectively. I am grateful to Gerhard Fischer for mentorship.

References

[1] S. Mendelsohn, Patterns formed by a single shot of malt, Information World Review (jun 2000).
URL http://www.iwr.co.uk/2082879

[2] C. Boulakia, Patent mapping, ScienceCareers.org (oct 2001).
URL http://sciencecareers.sciencemag.org/career_development/previous_issues/articles/1190/patent_mapping

[3] P. Rees, Patent pictures: It's patently good news, Research Information (jan/feb 2004).
URL http://www.researchinformation.info/rijanfeb04patents.html

[4] A. L. Porter, S. W. Cunningham, Tech mining: Exploiting New Technologies for Competitive Advantage, John Wiley & Sons, New York, 2005.

[5] C. J. van Rijsbergen, Information Retrieval, 2nd Edition, Butterworths, London, 1979.
URL http://www.dcs.gla.ac.uk/Keith/Preface.html

[6] H. P. Luhn, The automatic creation of literature abstracts, IBM Journal of Research and Development 2 (2) (1958) 159–165.
URL http://www.research.ibm.com/journal/rd/022/luhn.pdf

[7] G. K. Zipf, Human behavior and the principle of least effort, Addison-Wesley, Cambridge, MA, 1949.

[8] W. Li, Bibliography on Zipf's law (1997-2006).
URL http://www.nslij-genetics.org/wli/zipf/

[9] K. Papineni, Why inverse document frequency?, in: Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Pittsburgh, PA, 2001, pp. 1–8.
URL http://acl.ldc.upenn.edu/N/N01/N01-1004.pdf

[10] R. K. Belew, Finding Out About: Search Engine Technology from a Cognitive Perspective, Cambridge University Press, 2000.

[11] G. Salton, C.-S. Yang, On the specification of term values in automatic indexing, Journal of Documentation 29 (4) (1973) 351–372.

[12] K. Church, W. Gale, Inverse document frequency (idf): A measure of deviations from Poisson, in: D. Yarowsky, K. Church (Eds.), Proceedings of the Third Workshop on Very Large Corpora, Association for Computational Linguistics, Massachusetts Institute of Technology, Cambrige, MA, 1995, pp. 121–130.
URL http://www.aclweb.org/anthology/W95-0110

[13] G. Salton, C. Buckley, Term weighting approaches in automatic text retrieval, Information Processing and Management 24 (5) (1988) 513–523.

[14] A. J. Trippe, Patinformatics: Tasks to tools, World Patent Information 25 (3) (2003) 211–221.

[15] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill Book Company, New York, 1983.

[16] B. Vickery, A. Vickery, Information Science in Theory and Practice, Butterworths, London, 1987.

[17] A. J. Trippe, Visualization of chemical patents: Source titles and abstracts vs. enhanced titles and abstracts, in: 223rd American Chemical Society National Meeting, American Chemical Society, Orlando, FL, 2002.

[18] Thomson Scientific Professional Services, ThemeScape™ tips — Stopwords, in: Evaluation of IP — Using Aureka to Answer Key Business Questions, Aureka Master Class, London, 2006.

[19] M. P. Sinka, D. W. Corne, Evolving better stoplists for document clustering and web intelligence, in: A. Abraham, M. Köppen, K. Franke (Eds.), Design and Application of Hybrid Intelligent Systems, Vol. 104, IOS Press, Amsterdam, 2003, pp. 1015–1023.
URL http://www.sse.reading.ac.uk/common/publications/02065.pdf

[20] M. Fattori, G. Pedrazzi, R. Turra, Text mining applied to patent mapping: A practical business case, World Patent Information 25 (2003) 335–342.

[21] S. Robin, Statistical analysis of microarray data, Teaching material from Institut National Agronomique Paris-Grignon (may 2004).
URL http://www.inapg.fr/ens_rech/maths/Assets/bib/Microarray-tra.pdf

[22] A. J. Trippe, A comparison of ideologies: intellectually assigned co-coding clustering vs ThemeScape automatic thememtic mapping, in: Proceedings of the 2001 Chemical Information Conference, 2001.

[23] A. J. Trippe, Patinformatics: Identifiying haystacks from space, Searcher 10 (9) (2002) 28.

[24] G. Fischer, N. Lalyre, Analysis and visualisation with host-based software — The features of STN® AnaVist™, World Patent Information 28 (4) (2006) 312–318.

[25] R. T.-W. Lo, B. He, I. Ounis, Automatically building a stopword list for an information retrieval system, Journal of Digital Information Management 3 (1) (2005) 3–8.
URL http://ir.dcs.gla.ac.uk/terrier/publications/rtlo_DIRpaper.pdf

[26] S. Kullback, R. A. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1951) 79–86.

[27] R. K. Al-Halimi, F. W. Tompa, Using word position in documents for topic characterization, Tech. Rep. cs-2003-36, University of Waterloo, Canada (oct 2003).
URL http://www.cs.uwaterloo.ca/research/tr/2003/36/relevance_measures_TR.pdf