Formal proof of adequacy of document pre-processing in IF and IR.

Dominik Kuropka
Hasso Plattner Institute for IT-Systems Engineering
at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
dominik.kuropka@hpi.uni-potsdam.de

Abstract

In the context of Information Filtering and Retrieval the usage of pre-processing steps like Stopword-list, Stemming and Thesaurus Substitution is very common. Indeed the usage of those pre-processing steps seems to be sound at the first look; formal proofs for the adequacy of those steps are still absent. In this paper those proofs are given with the assistance of the Topic-based Vector Space Model. Additionally the hidden implicit assumptions of the above mentioned pre-processing steps are worked out and evaluated regarding their impact on practical usage.

1. Introduction

More and more unstructured documents are generated using computers today. The development of Internet supports this trend. Publishing documents is easier than ever before. The number of available documents is growing faster than ever. By now information has become one of the most important resources for business and research. But the fact that more information is available than ever before, doesn't necessarily lead to better decisions. Restricted capacity of human's information processing forces us to reduce the amount of information presented to the human. Today one of the greatest challenges is the efficient filtering and retrieval of information. While *Information Filtering* (IF) selects documents from a dynamic stream of documents using some kind of (static) profile, *Information Retrieval* (IR) selects relevant documents from a static set of documents based on a specified (ad hoc) query [BeCr1992]. Because of the high similarity of those tasks, most concepts and models for IF or IR can be adapted for both tasks [Kuro2004].

Witold Abramowicz (ed.), Business Information Systems, Proceedings of BIS 2005, Poznań, Poland

In today's literature a bunch of models and approaches is presented to solve this task in a more or less adequate and efficient manner. [BaRi1999; Kuro2004] When looking carefully at the different approaches it is striking that most approaches apply some pre-processing steps on documents. Usually these pre-processing steps are: Stopword-lists, Stemming and in some cases Thesaurus Substitution.

A *Stopword-list* (e.g. [Drott]) contains words, which are assumed to have no impact on the meaning of a document. Such a list usually contains words like 'the', 'is', 'a', etc. During the preprocessing all words matching the Stopword-list are removed from the document.

The term *Stemming* stands for the reduction of word forms e.g. 'houses', 'mice' to word stems e.g. 'hous', 'mic' (also known as Strong-Stemming) or to basic word forms e.g. 'house', 'mouse' (also known as Weak-Stemming). For pre-processing of English documents the PORTER stemming algorithm [Port80] is often used, other algorithms for example are: the Successor Variety Stemmer [HaWe74], the *n*-Gram Stemmer [AdBo74] and others like [Paic90]. In most IF and IR approaches documents are preprocessed by stemming algorithms. This results in the replacement of (theoretically) all word forms by their stem or basic word form. It is assumed that the reduction of different word forms to their stem or basic word form has no significant influence on the content of the document.

Finally, *Thesaurus Substitution* is defined as the replacement of different synonymous words by one leading word. For example the synonyms 'auto', 'automobile' and 'motor-car' can be replaced by the leading synonym 'car'. Which one of the synonyms is the leading synonym is usually defined in an arbitrarily manner. The usage of Thesaurus Substitution in document pre-processing is often justified by the observation that synonymous words have nearly the same meaning. From this follows that it does not matter which one of the synonyms is used in a document.

While many IF and IR approaches make use of Stopword-lists, Stemming and Thesaurus Substitution a formal proof of the adequacy of this kind of pre-processing is not given. Indeed, the usage of the above-presented pre-processing steps seems intuitively to be admissible, but this kind of argumentation is not an acceptable scientific procedure. In this paper, we will present a formal proof of adequacy of the usage of Stopword-lists, Stemming and Thesaurus Substitution on the base of the Topic-based Vector Space Model [BeKu2003] [Kuro2004] as explanation model. Additionally we will acquire the implicit assumptions, which arise from the usage of these pre-processing steps.

We will begin with a short introduction into the Topic-based Vector Space Model in section 2 and continue with three lemmas for our formal proof in section 3. Finally we present a summary in section 4.

2. TVSM at a glance

The Topic-based Vector Space Model (TVSM) [BeKu2003; Kuro2004] is a vector-based approach for document comparison in the context of IF and IR. The key features of this approach are the explicit assumption of dependences between terms, its flexibility regarding the specification of inter-term similarities and the integration of stopwords, stemming and some kind of thesaurus within the model. For our aim it is sufficient to narrow the introduction to the TVSM on the theoretical base of the model. A more detailed introduction regarding a possible implementation and linguistic details can be found in [BeKu2003] and [Kuro2004].

The fundamental assumption of the TVSM is the existence of a *d*-dimensional vector space *R*. Each axis intercept of this vector space may have only positive values[1] (including zero) and represents an elementary topic (e.g. literature, computer, medicine). (Hint: **R** is the set of real numbers, while **N** is the set of the natural numbers)

$$R = \mathbf{R}_{\geq 0}^{d} \text{ with } d \in \mathbf{N} \tag{1}$$

Further the TVSM assumes that terms (which are in this context equal to words) are the atomic elements of documents. The set *T* contains all possible terms, while a vector $\vec{t}_i$ represents the affiliation to the topics of a term *i* of the set *T*.

$$\forall i \in T: \quad \vec{t}_i \in R \quad \wedge \quad \left|\vec{t}_i\right| \in [0...1] \tag{2}$$

[1] The reason for having only positive axis intercepts is that a topic, which is for example represented by the word 'literature', cannot have some kind of reverse relationship to an other topic. In opposition to topics this is not true for word meanings. A word like 'warm' stands in a reverse relationship to 'cold'. But these words are attributes and not suitable for being a topic and they are highly dependent on the writer's point of view. A full discussion of this problem would go beyond the scope of this paper; this problem is addressed in detail in [Kuro2004].

The vector length $|\vec{t}_i|$ represents how good a term is able to indicate a topic in general. A low value means that the term is not significant at all (this usually holds for stopwords). Words that are strongly related to a topic are good indicators for the topic and have a vector length near one.

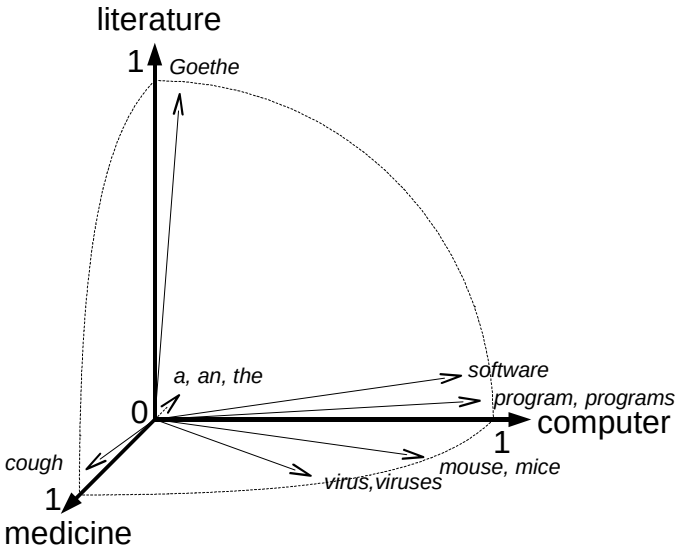

Figure 1: Interpretation of the TVSM vector space.

The whole circumstances of the representation of terms in the TVSM vector space are exemplified by Figure 1. In this simplified example the vector space has only three topic-dimensions: medicine, literature and computer. For example the term 'Goethe' is represented by a vector that points at the direction of the topic 'literature'. Whereas the terms 'virus' and 'viruses' point at a direction between the topics 'medicine' and 'computer' because these terms are related to both topics. Both terms ('virus' and 'viruses') have even the same vector due to they refer to the same topics. While 'Goethe' and 'virus' are represented by long vectors (vector length is near one) the vectors of the stopwords 'a', 'an' and 'the' are short (to be exact: the vector length of stopwords is null, which is for readability reasons not displayed in Figure 1), because these stopwords can not be related to any topic.

Figure 1 advises us to use the angle $\omega_{i,j}$ between two term vectors as an indicator for topic-related similarity between the two terms *i* and *j*. For normalization reasons we will define the similarity of two terms as the cosine of the angle between those terms:

$$\mathrm{sim}(i, j) = \cos \omega_{i,j} \in [0...1] \tag{3}$$

The angle mapping using the cosine function is unique and all results are positive in this case, because of the positive axis intercept constraint for the vector space. This constraint results in all angles being not larger than 90 degrees:

$$\omega_{i,j} \in [0°...90°] \tag{4}$$

We define *D* as the set of all documents. A document is represented in the TVSM as a weighted set of terms. A document vector $\vec{d}_k$ is assigned to each document *k* of the set *D*. The document vector is defined as follows:

$$\forall k \in D: \quad \vec{d}_k = \frac{1}{\left|\vec{\delta}_k\right|}\vec{\delta}_k \quad \Rightarrow \quad \left|\vec{d}_k\right| = 1 \tag{5}$$
$$\text{with } \vec{\delta}_k = \sum_{i \in T} a_{k,i}\vec{t}_t$$

and with $a_{k,i}$ being the occurrence number of term *i* in document *k* (also known as term frequency). This definition of document vectors standardizes all vectors to a vector length of value one. This standardization is derived from the assumption that all documents have some content and that a document cannot be without any relation to a topic, because in this case the document would consist only of stopwords. Documents consisting only of stopwords do not exist in common natural languages, because they have no meaning. This does not mean that a document must be related to exactly one or two topics. A document may even be related to all topics (which can be true for an index or broad overview document).

The topic related similarity sim(*k*,*l*) between two documents *k* and *l* is defined as the scalar product between the corresponding document vectors. Because the document vectors are standardized, the scalar product equals the cosine of the angle between the document vectors. Before we can compute the similarity between two documents we first have to compute the values for $\left|\vec{\delta}_k\right|$ and $\left|\vec{\delta}_l\right|$ respectively:

$$\begin{aligned} \left|\vec{\delta}_k\right| &= \left|\sum_{t\in T} a_{k,i}\vec{t}_i\right| \\ &= \sqrt{\left|\sum_{t\in T} a_{k,i}\vec{t}_i\right|^2} \\ &= \sqrt{\left(\sum_{t\in T} a_{k,i}\vec{t}_i\right)^2} \\ &= \sqrt{\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j} \end{aligned} \tag{6}$$

with $\vec{t}_i\vec{t}_j = \left|\vec{t}_i\right|\left|\vec{t}_j\right| \cdot \cos\omega_{i,j}$ being the scalar product between the term vectors of terms *i* and *j*.

Now the similarity between two documents can be computed as:

$$\begin{aligned} \mathrm{sim}(k,l) &= \vec{d}_k\vec{d}_l \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\vec{\delta}_k\vec{\delta}_l \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\sum_{i\in T} a_{k,i}\vec{t}_i\sum_{j\in T} a_{l,j}\vec{t}_j \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j \end{aligned} \tag{7}$$

This similarity definition provides values near zero for documents, which have different topics and values near one for document having the same topic or the same affiliation to several topics. Referring to Figure 1 a document containing the terms ‘software’, ‘programs’ and ‘mice’ would be very similar to a document containing ‘program’, ‘mouse’ and ‘virus’ because the vectors of all of these terms are oriented into the direction of the ‘computer’ topic-axis.

This description of the TVSM does not include an algorithm how terms should be assigned to topics. But the lemmas presented in the next section can be used to define some requirements on such an algorithm. A proposal for such an algorithm is presented in [Kuro2004].

3. Lemmas

This section presents three lemmas; any of them gives a formal proof of the adequacy for the usage of one of the in section 1 presented pre-processing steps. All proofs in this paper base on the TVSM as explanation model. Due to the formal approach the embedded assumptions, which are hidden in the pre-processing approaches, will come to the fore during proofing.

3.1. Stopword-Lemma

The Stopword-Lemma will proof the adequacy of the usage of Stopword-lists for the pre-processing of documents. This proof bases on the widely spread assumption that a stopword is a term without any kind of significance regarding a topic. Formally we have to define a set of all stopwords $T_\otimes$ as a subset of the set of all terms *T*. Because stopwords cannot be assigned to any topic, the term vectors of all stopwords have a length with the value zero:

$$\begin{aligned} & \left|\vec{t}_i\right| = 0 && \forall i \in T_\otimes \subset T \\ \Rightarrow \quad & \vec{t}_i\vec{t}_j = \left|\vec{t}_i\right|\left|\vec{t}_j\right|\cos\omega_{i,j} = 0 && \text{if } i \in T_\otimes \vee j \in T_\otimes \end{aligned} \tag{8}$$

From this assumption follows that the scalar product of two term vectors is always null if one of the terms is a stopword. Using this conclusion we can derive the similarity between two documents by only taking the non-stopwords into account. We can compute $\left|\vec{\delta}_k\right|$ and $\left|\vec{\delta}_l\right|$ respectively by merely using the non-stopwords. This fact can be derived from the following equation:

$$\begin{aligned} \left|\vec{\delta}_k\right| &= \sqrt{\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j} \\ &= \sqrt{\sum_{i\in T-T_\otimes}\sum_{j\in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{i\in T_\otimes}\sum_{j\in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}_{=0}} \\ &= \sqrt{\sum_{i\in T-T_\otimes}\left(\sum_{j\in T-T_\otimes} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{j\in T_\otimes} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}_{=0}\right)} \\ &= \sqrt{\sum_{i\in T-T_\otimes}\sum_{j\in T-T_\otimes} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j} \end{aligned} \tag{9}$$

The next equation shows that the similarity between two documents can be derived by using only the non-stopwords. This demonstrates, that the removal of stopwords as preprocessing step is an acceptable simplification if all stopwords have no relation to any topic.

$$\begin{aligned}\mathrm{sim}(k,l) &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\sum_{i\in T-T_\otimes}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{i\in T_\otimes}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j}_{=0} \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\sum_{i\in T-T_\otimes}\left(\sum_{j\in T-T_\otimes} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{j\in T_\otimes} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j}_{=0}\right) \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|}\sum_{i\in T-T_\otimes}\sum_{j\in T-T_\otimes} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j\end{aligned} \tag{10}$$

3.2. Stemming-Lemma

Using the Stemming-Lemma we will show under which assumptions the often-used reduction of words to their stem (strong-stemming) or basic word form (weak-stemming) is an adequate simplification procedure. To enhance readability we will abstract in this subsection from the two stemming possibilities and write only stemming as shortcut for strong- and weak-stemming and use the term ‘stem’ synonymously for stems in the strong-stemming case or for word forms in the weak stemming case. For the formal proof we have to make the following definitions: $T_\perp$ is the set of all stems and it is a subset of the set *T*. $\perp: T \to T_\perp$ is the stem assignment function and it assigns to each term the proper stem. The reverse relation is $\perp^{-1}: T_\perp \to \wp(T)$ assigns to all stems the set of possible word forms including the stem itself. Finally, we have to assume that all possible results of the reverse relation are disjoint, because otherwise the stem assignment function is not unique. Formally, all these definitions and assumptions look like this:

$$
\begin{array}{ll}
T_{\perp} \subseteq T & \\
\perp(i) \in T_{\perp} & \forall i \in T \\
\perp^{-1}(o) \subseteq T \wedge o \in \perp^{-1}(o) & \forall o \in T_{\perp} \\
\perp(i) = o & \forall o \in T_{\perp}, i \in \perp^{-1}(o) \\
\nexists i : i \in \perp^{-1}(o) \ \wedge \ i \in \perp^{-1}(p) & \forall o \neq p
\end{array}
\tag{11}
$$

A common assumption in the IF and IR context is that each word form has the same relations to a topic or a combination of topics as the stem of the word form. Formally this means that the term vectors of word forms and their stem point into the same direction and have the same length:

$$
\begin{array}{rl}
 & \omega_{i,o} = \omega_{0,i} = 0^{\circ} \quad \wedge \quad \left|\vec{t}_i\right| = \left|\vec{t}_o\right| \quad \forall i \in T, o = \perp(i) \\
\Rightarrow & \vec{t}_i = \vec{t}_o
\end{array}
\tag{12}
$$

This leads to the conclusion that the term vector of a stem and the vectors of the word forms of the stem are equal. Using this conclusion it is possible to compute $\left|\vec{\delta}_k\right|$ and $\left|\vec{\delta}_l\right|$ respectively by reducing all word forms to stems, as shown in the following equation:

$$
\begin{aligned}
\left|\vec{\delta}_k\right| &= \sqrt{\sum_{i \in T} \sum_{j \in T} a_{k,i} a_{k,j} \vec{t}_i \vec{t}_j} \\
&= \sqrt{\sum_{o \in T_{\perp}} \sum_{i \in \perp^{-1}(o)} \sum_{p \in T_{\perp}} \sum_{j \in \perp^{-1}(p)} a_{k,i} a_{k,j} \vec{t}_i \vec{t}_j} \\
&= \sqrt{\sum_{o \in T_{\perp}} \sum_{p \in T_{\perp}} \sum_{i \in \perp^{-1}(o)} \sum_{j \in \perp^{-1}(p)} a_{k,i} a_{k,j} \vec{t}_o \vec{t}_p} \\
&= \sqrt{\sum_{o \in T_{\perp}} \sum_{p \in T_{\perp}} \vec{t}_o \vec{t}_p \left(\sum_{i \in \perp^{-1}(o)} \sum_{j \in \perp^{-1}(p)} a_{k,i} a_{k,j} \right)} \\
&= \sqrt{\sum_{o \in T_{\perp}} \sum_{p \in T_{\perp}} \vec{t}_o \vec{t}_p \left(\sum_{i \in \perp^{-1}(o)} a_{k,i} \right) \left(\sum_{j \in \perp^{-1}(p)} a_{k,j} \right)} \\
&= \sqrt{\sum_{o \in T_{\perp}} \sum_{p \in T_{\perp}} a'_{k,o} \, a'_{k,j} \, \vec{t}_o \vec{t}_p} \\
\text{with } a'_{k,o} &= \sum_{i \in \perp^{-1}(o)} a_{k,i} \text{ and } a'_{k,p} = \sum_{j \in \perp^{-1}(p)} a_{k,j}
\end{aligned}
\tag{13}
$$

The reduction of all word forms to stems implies that the stem replaces all word forms and that the occurrence of the stem within the documents is

incremented by each replacement. Further, the similarity of documents can be computed in an analogous way $\left|\vec{\delta}_k\right|$ and $\left|\vec{\delta}_l\right|$ by only using stems and their incremented occurrences.

$$\begin{aligned} \mathrm{sim}(k,l) &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|} \sum_{i\in T}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j \\ &\vdots \\ &= \frac{1}{\left|\vec{\delta}_k\right|\left|\vec{\delta}_l\right|} \sum_{o\in T_\perp}\sum_{p\in T_\perp} a'_{k,o}\, a'_{l,p}\, \vec{t}_o\vec{t}_p \end{aligned} \tag{14}$$

$$\text{with } a'_{k,o} = \sum_{i\in\perp^{-1}(o)} a_{k,i} \text{ and } a'_{l,p} = \sum_{j\in\perp^{-1}(p)} a_{l,j}$$

This demonstrates the adequacy of stemming as a preprocessing step for IF and IR, if the following assumptions hold: First, the assignment of a stem to its topics must be equal to the assignment of all word forms of the stem to the topics. Second, the sets of word forms, which are assigned to two arbitrary stems, have to be disjoint. The last assumption does not hold for all words in every language. For example the German word form “sucht” can be the basic word form for “Sucht” (in English ‘addiction’) or the singular present form of “suchen” (in English ‘to search for’).

3.3. Synonymy-Lemma

This subsections shows under which circumstances the casually used Thesaurus Substitution is acceptable. You will notice that the Synonymy-Lemma is structurally equivalent to the Stemming-Lemma. The basic assumption of the Synonymy-Lemma is the existence of *total synonymy* between some words. Total synonymy means that two synonyms e.g. ‘car’ and ‘automobile’ mean the same, independently from the context they are used in. For the formalization we define T_f as the set of all leading terms (remember that leading terms can be defined in an arbitrarily manner), which is a subset of *T*, the set of all terms. Further, we define $F : T \to T_f$ as the leading term function. This function returns for every term the leading term, which is associated with the term. The reverse leading term function $F^{-1} : T_f \to \wp(T)$ returns for every leading term the set of all terms, which are associated with the leading term including the leading term itself. From the assumption of total synonymy we can derive that all result sets of F^{-1} must be disjoint for different input values. Formally we can write:

$$
\begin{array}{ll}
T_f \subseteq T & \\
F(i) \in T_f & \forall i \in T \\
F^{-1}(o) \subseteq T \wedge o \in F^{-1}(o) & \forall o \in T_f \\
F(i) = o & \forall o \in T_f, i \in F^{-1}(o) \\
\nexists i : i \in F^{-1}(o) \ \wedge \ i \in F^{-1}(p) & \forall o \neq p
\end{array} \tag{15}
$$

From the synonymy assumption also follows that the term vectors of different synonymous terms have to be equal. This means, they point into the same direction and have an equal length.

$$
\begin{array}{ll}
\omega_{i,o} = \omega_{0,i} = 0° \quad \wedge \quad \left|\vec{t}_i\right| = \left|\vec{t}_o\right| & \forall i \in T, o = F(i) \\
\Rightarrow \qquad \vec{t}_i = \vec{t}_o &
\end{array} \tag{16}
$$

Having this formal munition, we can derive $\left|\vec{\delta}_k\right|$ and $\left|\vec{\delta}_l\right|$ respectively by the use of the leading terms, similarly to the Stemming-Lemma.

$$
\begin{aligned}
\left|\vec{\delta}_k\right| &= \sqrt{\sum_{i \in T} \sum_{j \in T} a_{k,i} a_{k,j} \vec{t}_i \vec{t}_j} \\
&\vdots \\
&= \sqrt{\sum_{o \in T_f} \sum_{p \in T_f} a'_{k,o} \, a'_{k,p} \, \vec{t}_o \vec{t}_p} \\
\text{with } a'_{k,o} &= \sum_{i \in F^{-1}(o)} a_{k,i} \text{ and } a'_{k,p} = \sum_{j \in F^{-1}(p)} a_{k,j}
\end{aligned} \tag{17}
$$

The substitution of synonymous terms by a leading term implies the increment of the occurrence of the leading term for each done substitution. The similarity of documents can be derived with the use of leading terms by the following equation:

$$
\begin{aligned}
\text{sim}(k,l) &= \frac{1}{\left|\vec{\delta}_k\right| \left|\vec{\delta}_l\right|} \sum_{i \in T} \sum_{j \in T} a_{k,i} a_{l,j} \vec{t}_i \vec{t}_j \\
&\vdots \\
&= \frac{1}{\left|\vec{\delta}_k\right| \left|\vec{\delta}_l\right|} \sum_{o \in T_f} \sum_{p \in T_f} a'_{k,o} \, a'_{l,p} \, \vec{t}_o \vec{t}_p \\
\text{with } a'_{k,o} &= \sum_{i \in F^{-1}(o)} a_{k,i} \text{ and } a'_{l,p} = \sum_{j \in F^{-1}(p)} a_{l,j}
\end{aligned} \tag{18}
$$

The above equation shows that the use of Thesaurus-Substitution is theoretically possible, if the assumption of total synonymy is true. But in fact, total synonymy is an artificial construct for most natural languages. Rather we can observe that most common synonymous words have only in some contexts the same meaning and in other contexts a different or no meaning. For example, the words 'rock' and 'stone' have in most contexts a synonymous meaning. But in the context of fruits 'stone' has the meaning of the hard seed, which is inside of some fruits, while the word 'rock' is not used in this context. This kind of synonymy is known as *partial synonymy*. If 'stone' is substituted by 'rock' (or vice versa) some information may got lost in the context of fruits. Especially two technical terms may have a synonymous meaning in common language while they have different meanings in some special fields. (E.g. 'costs' and 'expenses' have usually a synonymous meaning in common language while they have a different meaning in the field of business administration. In this field 'costs' are burdens, which result immediately from the usage of something, but which do not necessarily cause a payment. While 'expenses' are payments which do not necessarily are the result of an event or the usage of something.)

So we see that the usage of Thesaurus Substitution may be problematic due to the embedded artificial assumption of total synonymy, which is covered within this approach.

4. Summary

This paper gives a formal proof of the adequacy of the usage of the widely used preprocessing steps in the context of IF and IR. These steps are namely: Stopword-list, Stemming and Thesaurus Substitution. The Topic-based Vector Space Model is used for the proof as an explanation model. Additionally the implicit assumptions of the usage of those preprocessing steps are presented. The result of this paper can be summarized as follows: The usage of a Stopword-list is an acceptable simplification due to the only implicit assumption is that all stopwords have no relation to any topic. This assumption can be met easily. The usage of Stemming is acceptable in most cases, although the implicit assumption of disjoint word form sets, which are assigned to each stem, cannot be held in some cases. The number of those cases is dependent on the processed language. Finally the usage of Thesaurus Substitution has to be evaluated as problematic due to its implicit total synonymy assumption. The construct of total synonymy is artificial for most natural languages. This fact makes the usage of Thesaurus Substitution problematic especially for document, which have a broad topic spectrum.

It is also mentionable that the Topic-based Vector Space Model is not only suitable for a direct appliance in IF and IR, but it is also suitable as an explanation model for the formal deduction of new insights in the IF and IR context. The formal adequacy of usual preprocessing steps puts the application of them on TVSM based IF and IR methods on a well grounded base. This is a feature of the TVSM which is not meet by most other IF and IR models like the Vector Space Model [Salt1968], the Binary Independence Retrieval [RoJo1976], the Generalized Vector Space Model [WZRW1987] and the Latent Semantic Index model [FDDL1988].

5. References

1. [AdBo74] G. Adamson, J. Boreham: *The Use of an Association Measure Based on Character Structure to Identify Semantically Related Pairs of Words and Document Titles*. In: Information Storage and Retrieval (10), 1974, pp. 253-260.
2. [BaRi1999] R. Baeza-Yates, B. Ribeiro-Neto: *Modern Information Retrieval*. Addision Wesley Publishing Company, 1999.
3. [BeCr1992] N. J. Belkin, W. B. Croft: *Information Filtering and Information Retrieval: Two Sides of the Same Coin?* In: Communications of the ACM, 35(12), 1992, pp. 29-38.
4. [BeKu2003] J. Becker, D. Kuropka: *Topic-based Vector Space Model*. In: Proceedings of the 6th International Conference on Business Information Systems, 2003, pp. 7-12.
5. [Drott] M. Drott: *A big Stop List*. http://drott.cis.drexel.edu/retrieval.html
6. [HaWe74] M. Hafer, S. Weiss: *Word Segmentation by Letter Successor Varieties*. In: Information Storage and Retrieval (10), 1974, pp. 371-385.
7. [Kuro2004] D. Kuropka: *Modelle zur Repräsentation natürlichsprachlicher Dokumente – Information-Filtering und -Retrieval mit relationalen Datenbanken*, Logos, Berlin, 2004. (in German)
8. [Paic90] C. Paice: *Another Stemmer*. In ACM SIGIR Forum 24(3), 1990, pp. 56-61.
9. [Port80] M. Porter: *An algorithm for suffix stripping*. In: Program 14(3), 1980, pp. 130-137.
10. [Salt1968] G. Salton: Automatic Information Organization and Retrieval. McGraw-Hill, New York, 1968.
11. [RoJo1976] S. E. Robertson, K. S. Jones: Relevance weighting of search terms. In: Journal of the American Society for Information Sciences, 27(3), 1976, pp. 129-146.
12. [WZRW1987] S. K. Wong, W. Ziarko, V. V. Raghaven, R. C. N. Wong: On Modeling of Information Retrieval Concepts in Vector Spaces. In: ACM Transactions on Database Systems, 12(2), 1987, pp. 299-321.
13. [FDDL1988] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, et al.: Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure. In: Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1988, pp. 465-480.