computation

MDPI

Article

Tensor-Based Semantically-Aware Topic Clustering of Biomedical Documents

Georgios Drakopoulos [1], Andreas Kanavos [2,*], Ioannis Karydis [1], Spyros Sioutas [1] and Aristidis G. Vrahatis [2]

[1] Department of Informatics, Ionian University, Tsirigoti Square 7, 49100 Kerkyra, Greece; c16drak@ionio.gr (G.D.); karydis@ionio.gr (I.K.); sioutas@ionio.gr (S.S.)
[2] Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece; agvrahatis@upatras.gr
* Correspondence: kanavos@ceid.upatras.gr

Received: 28 April 2017; Accepted: 13 July 2017; Published: 18 July 2017

Abstract: Biomedicine is a pillar of the collective, scientific effort of human self-discovery, as well as a major source of humanistic data codified primarily in biomedical documents. Despite their rigid structure, maintaining and updating a considerably-sized collection of such documents is a task of overwhelming complexity mandating efficient information retrieval for the purpose of the integration of clustering schemes. The latter should work natively with inherently multidimensional data and higher order interdependencies. Additionally, past experience indicates that clustering should be semantically enhanced. Tensor algebra is the key to extending the current term-document model to more dimensions. In this article, an alternative keyword-term-document strategy, based on scientometric observations that keywords typically possess more expressive power than ordinary text terms, whose algorithmic cornerstones are third order tensors and MeSH ontological functions, is proposed. This strategy has been compared against a baseline using two different biomedical datasets, the TREC (Text REtrieval Conference) genomics benchmark and a large custom set of cognitive science articles from PubMed.

Keywords: humanistic data; higher order data; medical information retrieval; topic clustering; PubMed; MeSH Ontology; tensor algebra; tucker factorization

1. Introduction

Cognitive science, namely the study of the mind and its processes [1–3], has recently gained significant momentum, which can be attributed to a number of reasons. It is a major driver of the big data age along with online social networks, the semantic web and computational systems theory to name a few. Recent sociological and demographic studies conducted in the majority of the Western world including the EU [4] and the U.S. [5] reveal that one of the biggest challenges of the next decade will be the healthcare costs associated with cognitive issues. Similar trends at the planet scale can be found in reports compiled by the UN Population Division [6–8]. Thus, the systematic analysis of cognitive science literature is of immediate interest, besides researchers, of healthcare planners, government agencies, hospital administrators, insurance companies, equipment manufacturers and software developers.

With the creation of PubMed (http://www.ncbi.nlm.nih.gov/pubmed) in 1996, the largest public online database under the ultimate administrative oversight of NIH, a massive collection spanning millions of life science articles is available to researchers. Over the past two decades, it has been substantially enriched and currently contains more than fourteen million abstracts, whereas it accepts and serves more than seventy million queries of six terms each on average per month. Indicative of its enormous topic diversity is the fact that through Entrez (French term for enter.), the PubMed-coupled

indexing engine, the searchable keywords currently exceed two millions. Two issues arising in any information retrieval context, especially in a major digital repository of humanistic data, are precision and recall. The former pertains to the accurate retrieval of studies, articles and data tables stored across a variety of online archives and libraries that are relevant to the query, whereas the latter refers to the fraction of relevant documents that is retrieved for a given query [9].

PubMed cognitive science-related documents range over such diverse types as academic articles, software blueprints, demographic surveys, healthcare personnel training manuals, best practice guides and clinical data reports. It is evident that in such a large collection, both in terms of quantity and quality, information retrieval and text mining techniques should be applied if any meaningful piece of information is to be acquired. This is especially true given that medical and scientific articles exhibit a high degree of textual structure, including metadata like abstracts and keywords.

The primary contribution of this article is a topic-based clustering strategy for cognitive documents whose core is a third order term-keyword-document tensor. The latter is one of the many possible direct generalizations of the established term-document matrix, which essentially is a second order tensor from a linear algebraic point of view. As such, this tensor can be queried in a similar manner. The reason for selecting this particular scheme over a number of other possible ones stems from the intuition, corroborated in part from empirical scientometric evidence, that keywords are semantically more important compared to ordinary terms [10,11].

This journal article is structured as follows. Section 2 summarizes recent work on medical retrieval and tensor analysis. Section 3 describes software tools, and Section 4 outlines the proposed tensor model. Furthermore, Section 5 discusses performance aspects of the proposed and the baseline method obtained through the TREC (Text REtrieval Conference) genomics dataset, while Section 6 discusses tensor analytics. Finally, Section 7 explores future research directions. Tensors are denoted by uppercase calligraphic letters, such as $\mathcal{T}$, matrices by uppercase boldface letters, like $\mathbf{M}$, and vectors by lowercase boldface letters or numbers, like $\mathbf{v}$ and $\mathbf{1}$. Table 1 summarizes the article notation.

Table 1. Article notation.

Symbol	Meaning
$\triangleq$	Definition or equality by definition
$\{x_k\}$	Set consisting of elements x_k
$\lvert S \rvert$	Cardinality of set S
$\langle x_k \rangle$	Sequence with elements x_k
$\times$	Cartesian product
$\times_k$	Tensor product along k-th dimension
$\lVert \cdot \rVert_F$	Frobenius tensor or matrix norm
$\mathbf{1}_n$	Vector with n entries of 1
$Q(p)$	Indicator function for predicate p
$\mathcal{H}(s_1, \ldots, s_n)$	Harmonic mean of values $s_1, \ldots, s_n$
$\mathrm{E}[X]$	Expected value of random variable X
$\mathrm{Var}[X]$	Variance of random variable X
$\kappa_3[X]$	Skewness of random variable X
$\kappa_4[X]$	Kurtosis of random variable X

2. Previous Work

Document clustering has gained much interest in biomedicine [12]. PubMed abstracts are clustered with frequent words and near terms in [13]. A graph algorithm based on flow simulation is considered in [14], where advanced techniques are proposed in [15].

Biomedical ontologies in conjunction with mining of biomedical texts led to the technique of word sense disambiguation (WSD), which maps documents to different topics. Ontologies and meta-data assist the clustering algorithms [16]. Event-based text mining systems in the context of

biomedicine as an annotation scheme are the focus of [17]. On the other hand, domain-specific information extraction systems regarding event-level information with automatic causality recognition are proposed in [18]. Human gene ontologies are described in [19,20]. U-Compare, an integrated text mining and NLP system based on the Unstructured Information Management (UIMA) Framework (UIMA: http://uima.apache.org/), is presented in [21].

Using the MeSH ontology for biomedical document clustering is popular in scientific literature [22–25]. Various clustering approaches such as suffix tree clustering were supplemented with ontological information in [26], whereas the accuracy of similarity metrics is discussed in [27]. A knowledge domain scheme based on bipartite graphs with MeSH is presented in [28]. Two serious limitations that face approaches by using the MeSH thesaurus are introduced in [29].

Tensor algebra [30,31] and the closely-associated field of multilayer graphs [32,33] are some of the primary algorithmic tools for dealing with higher order data, along with higher order statistics [34,35] and multivariate polynomials [36,37]. Central places in tensor algebra have Tucker and Kruskal tensor forms [38], which allow alternative tensor representations appropriate for certain linear algebraic operations such as tensor-matrix multiplication, tensor compression [39], tensor regularization and factor discovery. Models for tensor data mining have been outlaid in [40]. A very recent work combining tensors and semantics for medical information retrieval is [41].

3. System

3.1. Architecture

The proposed system architecture is shown in Figure 1. The interaction between the various components has been kept at a minimum, and feedback loops have been avoided. However, in future versions, the tensor can be updated either incrementally or in batch mode with information extracted from the queries.

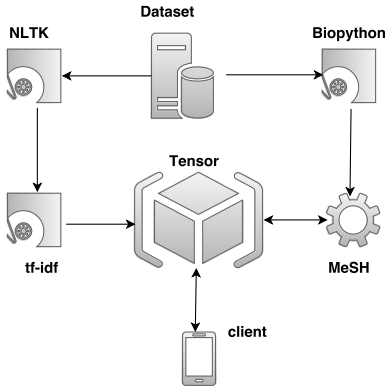

Figure 1. System architecture.

3.2. Python Tools

Python is well known in the developer community for its rich library ecosystem.

The objective of the Entrez document retrieval system is to provide a single entry point for seamless and efficient access across those health-related public databases that are under NIH administrative supervision, including among others, PubMed, MEDLINE, preMEDLINE and the NCBI database. Thus, Entrez is the key to a vast body of medical knowledge through advanced text queries. As a consequence, APIs for Entrez have been implemented for most, if not all, major programming languages such as the NCBI API for Java, the NCBI Toolkit for C++ and Biopython for Python. The functionality of each Entrez API and the associated library should include at least methods for retrieving articles based on keywords, terms, authors, doi (Digital Object Identifier) or the

unique PubMed identifier, as well as for providing pointers to related documents or supplementary data and traversing document lists in both directions.

The native document format supported by Entrez is XML (ftp://ftp.ncbi.nlm.nih.gov/bioproject). The latter being structurally balanced and semantically enriched with tags and properties and possessing a strict tree hierarchy is particularly suited to parsing techniques such as those found in the Xerces family of Java parsers. Moreover, the highly structured XML format is appropriate for graph databases, such as TitanDB and Neo4j (https://neo4j.com) or, with appropriate conversion to JSON, for document databases, such as MongoDB. Table 2 contains the XML tags described in the public Entrez XML schema. An XML schema is one of the two means for formatting an XML document in a tree structure, the other one being DTD (Document Type Definition) [42]. Generally, a schema is preferred because of its increased flexibility, being itself written in XML. In contrast, DTD is based on a terse and restricted SGML syntax, which provides compatibility with the SGML standard at the expense of a steeper learning curve [43].

Table 2. PubMed document XML tags.

Abstract	CollectiveName	GroupList	LastName	PubDate
Affiliation	CopyrightInfo	History	LastPage	PublicationType
Article	Day	Identifier	MiddleName	PublisherName
ArticleId	ELocationID	IndividualName	Month	Replaces
ArticleIdList	FileHeader	ISSN	Number	Season
ArticleSet	FirstName	Issue	Object	Suffix
ArticleTitle	FistPage	Journal	ObjectList	VernacularTitle
Author	Group	JournalTitle	OtherAbstract	Volume
AuthorList	GroupName	Language	Param	Year

Biopython is one such PubMed API aiming at providing seamless and fully-fledged Entrez functionality, including document retrieval in a multitude of ways. Table 3 summarizes the methods that are associated with the basic Entrez functionality.

Table 3. Biopython methods.

Method	Task	Method	Task
efetch	Retrieves records from an id list	esummary	Finds document summaries from an id list
epost	Posts a file containing an id list	egquery	Provides Entrez database counts in XML
esearch	Searches and retrieves an id list	espell	Retrieves spelling suggestions
elink	Gets external articles from an id list	eread	Obtains the XML tree from Entrez
einfo	Provides fields for each database	parse	Parses the XML tree
close	Terminates established connection	read	Returns handler data

Once Biopython has been installed through pip or another Python package manager, it can be invoked as follows:

```
>>> from Bio import Entrez
>>> Entrez.email = 'name@domain.org'
>>> EntryPoint = Entrez.einfo()
>>> XMLArticle = EntryPoint.eread()
>>> EntryPoint.close()
```

Key ontological MeSH operations such as search and least common ancestor location can be automated with NLTK (http://www.nltk.org), a common library for natural language processing. Moreover, NLTK has been integrated with additional functionality for word- and sentence-level syntactic analysis, term similarity metrics, including the Wu–Palmer [44], the Leacock–Chodorow [45]

and the Jiang–Conrath [46] metrics, and methods for sub-thesaurus construction and maintenance. For instance, using NLTK and the term cognitive, the entries of Table 4 were located in the MeSH ontology.

Table 4. Cognitive-related MeSH entries.

ID	Entry	ID	Entry
F02.463.188.305	Cognitive dissonance	F02.463.188.331	Cognitive reserve
F03.615	Neurocognitive disorders	F03.615.250.700	Cognitive dysfunction
F04.096.628.255	Cognitive science	F04.096.628.255.500	Cognitive neuroscience
F04.754.137.365	Cognitive remediation	F04.754.137.428	Cognitive therapy
G07.345.124.260	Cognitive aging	H01.158.610.030	Cognitive neuroscience

Notice that the entries of Table 4 are located at very different levels of the MeSH tree hierarchy ranging from a high level of abstraction such as F03.615 down to very specialized issues like F04.096.628.255.500. Thus, subsequent searches started at high abstraction levels such as F02.463 and H01.158, which were identified by pruning the MeSH identifiers to their first two segments.

The following code segment displays how NLTK can parse a simple sentence.

```
>>> import nltk as nl
>>> from stemming.porter2 import stem
>>> from nl.corpus import stopwords
>>> print nl.word_tokenize('Hello␣world!')
```

3.3. Tensor Toolbox

Tensor Toolbox is a recent MATLAB toolbox by Sanida Labs for direct support of tensors and certain associated key functions [30]. Although MATLAB inherently supports multidimensional arrays since its earliest editions, Tensor Toolbox offers considerably more flexibility, a set of new and equivalent tensor types, including natural, compressed, Tucker, and Kruskal forms, and a broad set of methods for handling these primary data types. These primary data types are respectively denoted as tensor, sp tensor, t tensor and k tensor, constituting an important semantic difference compared to the default MATLAB approach, which treats all multidimensional arrays as ordinary matrix types. Since tensors represent natural keyword-term-document triplets, Tensor Toolbox is an indispensable software component of our implementation. Provided the Tensor and the Communications Systems toolboxes have been properly installed, the following MATLAB commands populate a sparse third order tensor and store it in Tucker form:

```
>> rng shuffle
>> for k = 1:4
>>   T(:,:,k) = randsrc(4, 8, [-1:1; 0.2 0.6 0.2]);
>> end
>> T = mat2ten(T);
>> TT = ttensor(T);
```

4. Proposed Model

4.1. Representation

Definition 1. *A p-th order tensor $\mathcal{T} \in \mathbb{S}_1 \times \mathbb{S}_2 \ldots \times \mathbb{S}_p$ is a p-dimensional array indexed by p integers and coupling simultaneously at most p distinct linear spaces denoted by $\mathbb{S}_k$, $1 \leq k \leq p$. When each of the p linear spaces $\mathbb{S}_k$ is $\mathbb{R}^{I_k}$, then as a shorthand $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_p}$.*

It is obvious from the way tensors are defined that they are direct generalizations of matrices. Indeed, a matrix $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$ is the linear algebraic vehicle for coupling the row space $\mathbb{R}^{I_1}$ and the column space $\mathbb{R}^{I_2}$. Of course, for square and invertible matrices, the row and column spaces coincide.

Tensor $\mathcal{G}$, which will contain properly-defined values for the keyword-term-document triplets, is populated by I_k keywords and I_t terms stored in I_d documents making it a third order tensor $\mathcal{G} \in \mathbb{R}^{I_k \times I_t \times I_d}$. Note that the proposed $\mathcal{G}$ is but one of the ways for extending the established term-document model, which is based on a second order tensor, namely a matrix $\mathbf{M} \in \mathbb{R}^{I_t \times I_d}$. For instance, in [47], a term-author-document is proposed. The latter is based on empirical scientometric evidence in favor of the semantic role authors play in the process of information retrieval [10,11]. A common point with the proposed model, besides both relying on third order tensors, is that they are inspired by OLAP (Online Analytical Processing) cubes [48]. Regarding the tensor dimensions, it should be noted that, although the three dimensions are easy to visualize and handle, they are by no means a golden rule.

As stated earlier, $\mathcal{G} \in \mathbb{R}^{I_k \times I_t \times I_d}$, essentially the algebraic cornerstone of the proposed technique, is a third order and real valued tensor simultaneously coupling the keyword, term and document spaces. The entries of $\mathcal{G}$ are associated with the document retrieval process. Concretely, let $k[i_1]$, $t[i_2]$ and $d[i_3]$ respectively denote the i_1-th keyword, the i_2-th term and the i_3-th document where $1 \leq i_1 \leq I_t$, $1 \leq i_2 \leq I_k$ and $1 \leq i_3 \leq I_d$. Moreover, let $f_k[i_1, i_3]$ and $f_t[i_2, i_3]$ respectively be the number of occurrences of $k[i_1]$ and $t[i_2]$ in $d[i_3]$. Then, $\mathcal{G}[i_1, i_2, i_3]$ contains the normalized occurrences of $k[i_1]$ and $t[i_2]$ in $d[i_3]$ according to the following four factor double tf-idf (term frequency-inverse document frequency) scheme:

$$\mathcal{G}[i_1, i_2, i_3] \triangleq p_k[i_1, i_3]\, q_k[i_1, i_3]\, p_t[i_2, i_3]\, q_t[i_2, i_3] \;=\; \text{tfidf}[i_1, i_3]\, \text{tfidf}[i_2, i_3] \tag{1}$$

In Equation (1), the first pair of terms $p_k[i_1, i_3]$ and $q_k[i_1, i_3]$ forms a standard tf-idf scheme based only on terms and documents:

$$\begin{aligned} p_k[i_1, i_3] &\triangleq Q(k[i_1] \in d[i_3])\, (1 + \log\,(1 + f_k[i_1, i_3])) \\ q_k[i_1, i_3] &\triangleq Q(k[i_1] \in d[i_3])\, \log \frac{I_d}{1 + \sum_{j=1}^{I_d} Q(k[i_1] \in d[j])} \end{aligned} \tag{2}$$

while the second pair of terms $p_t[i_2, i_3]$ and $q_t[i_2, i_3]$ constitutes the second tf-if scheme:

$$\begin{aligned} p_t[i_2, i_3] &\triangleq Q(t[i_2] \in d[i_3])\, (1 + \log\,(1 + f_t[i_2, i_3])) \\ q_t[i_2, i_3] &\triangleq Q(t[i_2] \in d[i_3])\, \log \frac{I_d}{1 + \sum_{j=1}^{I_d} Q(t[i_2] \in d[j])} \end{aligned} \tag{3}$$

4.2. Analytics

Tensor density, similarly to large matrix sparsity, is a significant metric, which besides potential compression, may reveal interesting patterns along many dimensions since its definition is straightforward.

Definition 2. *The density ρ of a tensor $\mathcal{T}$ is defined as the number of the non-zero elements to its total number of elements, which can be easily found by multiplying the size of each dimension. Thus:*

$$\rho \triangleq \frac{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_p=1}^{I_p} Q(\mathcal{T}[i_1, i_2, \ldots, i_p] \;\neq\; 0)}{\prod_{k=1}^{p} I_k} \tag{4}$$

Definition 3. *Along similar lines, the log density ρ' of $\mathcal{T}$ is defined as the logarithm of the number of the non-zero elements to the logarithm of its total number of elements, essentially being the ratio of the magnitudes of the respective numbers.*

$$\begin{aligned} \rho' &\triangleq \frac{\log\left(\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_p=1}^{I_p} Q\left(\mathcal{T}[i_1,i_2,\ldots,i_p]\neq 0\right)\right)}{\log\left(\prod_{k=1}^{p} I_k\right)} \\ &= \frac{\log\left(\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_p=1}^{I_p} Q\left(\mathcal{T}[i_1,i_2,\ldots,i_p]\neq 0\right)\right)}{\sum_{k=1}^{p}\log I_k} \end{aligned} \tag{5}$$

Besides its natural interpretation, ρ' can usually lead to larger values, which in turn result in numerically stable computations in formulae when it appears in denominators.

The Frobenius norm of a tensor $\mathcal{T}$, denoted by $\|\mathcal{T}\|_F$, is an algebraic indicator of the overall strength of the tensor entries, which is indirectly tied to compressionability. Recall that the Frobenius norm for a matrix $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$ is defined as:

$$\|\mathbf{M}\|_F \triangleq \operatorname{tr}\left(\mathbf{M}^T\mathbf{M}\right) = \left(\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\mathbf{M}^2[i_1,i_2]\right)^{\frac{1}{2}} \tag{6}$$

Both are related in their own way to compression potential, which is critical given the large volume of data typically held in tensors. The former plays the same role as with matrices, whereas the latter indicates whether there are strong or weak connections between keywords, terms and documents. Since both provide a data summary in the form of a scalar, they give quick and overall information regarding the tensor status at the expense of aggregating information about each dimension of this single value. Thus, both metrics can be used as building blocks for composite ones, which examine each dimension separately.

Definition 4. *The Frobenius norm $\|\mathcal{T}\|_F$ of an p-th order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_p}$ is the square root of the sum along each dimension of its elements squared:*

$$\|\mathcal{T}\|_F \triangleq \left(\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_p=1}^{I_p}\mathcal{T}i^2[i_1,i_2,\ldots,i_p]\right)^{\frac{1}{2}} = \left(\sum_{i_1,\ldots,i_p}\mathcal{T}^2[i_1,\ldots,i_p]\right)^{\frac{1}{2}} \tag{7}$$

Generally, there is no consensus as to which values of $\|\mathcal{T}\|_F$ indicate strong connections on average. In order to derive bounds, probabilistic techniques can be employed by treating the elements of $\mathcal{T}$ being drawn from a distribution. One way is to observe that $\|\mathcal{T}\|_F^2$ is the sample approximation of $\mathrm{E}\left[\mathcal{T}^2[i_1,\ldots,i_p]\right]$. Then, since the Frobenius norm is always positive, as all-zero tensors are not under consideration, the Markov inequality:

$$\operatorname{prob}\{X \geq \tau_0\} \leq \frac{\mathrm{E}[X]}{\tau_0}, \qquad X, \tau_0 > 0 \tag{8}$$

can be used to derive a bound. For instance, if the elements of $\mathcal{T}$ are drawn from a Gauss distribution, then $\|\mathcal{T}\|_F$ follows a noncentral chi square distribution.

4.3. Metric Fusion

Again, in order to take into account information about each dimension separately for a third order tensor, it suffices to fix the last index and create a metric ν that takes into consideration the

density of each resulting matrix separately. Thus, if the density of each separate matrix $\mathcal{T}[:,:,i_3]$ for a fixed value of i_3 is defined as:

$$\rho_{i_3} \triangleq \frac{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} Q(\mathcal{T}[i_1,i_2,i_3] \neq 0)}{I_1 I_2} \tag{9}$$

then the set of tensor densities along the third dimension $\{\rho_{i_3}\}$ can be used to build the following aggregative metric:

$$\nu \triangleq \mathcal{H}\left(\rho_1, \ldots, \rho_{I_3}\right) = \frac{I_3}{\sum_{i_3=1}^{I_3} \frac{1}{\rho_{i_3}}} \tag{10}$$

The harmonic mean ensures that ν will tend to be close to the smallest of ρ_{i_3}.

For a third order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, a related metric can be constructed by first fixing one of the three indices, treating the remaining two dimensions as a sequence of matrices, computing the Frobenius norm for each such matrix and taking the harmonic mean of these norms. Deciding which index is to be fixed is important as it essentially determines a tensor partitioning. For the purposes of this article, the last index i_3 is fixed creating thus a metric μ, which ranges over the documents.

$$\mu \triangleq \mathcal{H}\left(\|\mathcal{T}[:,:,i_1]\|_F, \ldots, \left\|\mathcal{T}\left[:,:,i_p\right]\right\|_F\right) = \frac{I_3}{\sum_{i_3=1}^{I_3} \frac{1}{\|\mathcal{T}[:,:,i_3]\|_F}}, \quad \|\mathcal{T}[:,:,i_3]\|_F \neq 0 \tag{11}$$

Notice that $\mathcal{T}[:,:,k]$ in (11) denotes the matrix created by fixing i_3 to k while the two remaining indices stay unaltered, creating thus a $I_1 \times I_2$ matrix. If I_3 is large, which may well be the case for document collections, then it would also make sense to compute statistic measures such as sample versions of variance, skewness and kurtosis.

$$\begin{aligned} \kappa_3[X] &= \frac{\mathrm{E}\left[X^3\right]}{\mathrm{Var}[X]^{\frac{3}{2}}} \\ \kappa_4[X] &= \frac{\mathrm{E}\left[X^4\right]}{\mathrm{Var}[X]^2} \end{aligned} \tag{12}$$

4.4. Tensor Tucker Form

Definition 5. *The multiplication $\mathcal{T} \times_k \mathbf{v}$ along the k-th dimension between a p-th order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_k \times \ldots \times I_p}$ and a vector $\mathbf{v} \in \mathbb{R}^{I_k}$ is a $(p-1)$-th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_{k-1} \times I_{k+1} \times \ldots \times I_p}$ with elements:*

$$\mathcal{A}[i_1, \ldots i_{k-1}, i_k, \ldots, i_p] \triangleq \sum_{i_k=1}^{I_k} \mathcal{T}[i_1, \ldots, i_p]\, \mathbf{v}[i_k] \tag{13}$$

Definition 6. *The multiplication $\mathcal{T} \times_k \mathbf{M}$ along the k-th dimension between a p-th order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_k \times \ldots \times I_p}$ and a matrix $\mathbf{M} \in \mathbb{R}^{L \times I_k}$ is a p-th order tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots I_{k-1} \times L \times I_{k+1} \ldots \times I_p}$ with elements:*

$$\mathcal{B}[i_1, \ldots \ell, \ldots, i_p] \triangleq \sum_{i_k=1}^{I_k} \mathcal{T}[i_1, \ldots, i_p]\, \mathbf{M}[\ell, i_k] \tag{14}$$

Definition 7. *Tucker tensor factorization is defined as:*

$$\mathcal{T} = \mathcal{K} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_p \mathbf{U}_p = \mathcal{K} \prod_{k=1}^{p} \times_k \mathbf{U}_k \tag{15}$$

The Tucker factorization is one of the possible generalizations of the SVD for matrices:

$$\mathbf{M} = \mathbf{U}_1 \mathbf{K} \mathbf{U}_2^T = \mathbf{K} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \tag{16}$$

which is the core of the term-document information retrieval model and the starting point of a number of document clustering schemes. In order to compute the Tucker factorization, the higher order SVD is employed. The latter is based on cyclically updating each of the basis matrices $\mathbf{U}_k$ until they all converge according to a criterion.

4.5. Queries

Similarly to the term-document matrix case, tensor $\mathcal{G}$ can be queried regarding a set of terms $\{k[i_1]\}$ or a set of keywords $\{t[i_2]\}$. Said queries can be cached in terms of linear algebra as tensor-vector multiplications. In addition, $\mathcal{G}$ allows queries about both terms and keywords.

Generating a query vector, it suffices to place one at a position corresponding to a query term and zero otherwise. For a more detailed description, see the collection querying algorithm in [47].

4.6. Document Clustering

When partitioning a set S into k subsets, a heuristic approach is necessary since the number of ways b_k to perform such partitioning equals [49]:

$$b_{k+1} = \sum_{j=0}^{k} \binom{k}{j} b_j \quad 0 \leq k \leq |S|, \quad b_0 = 1, b_1 = 1 \tag{17}$$

The generating function $B(z)$ of the recursively defined sequence $\langle b_k \rangle$ is:

$$B(z) \triangleq \sum_{k=0}^{+\infty} b_k \frac{z^k}{k!} = \sum_{k=0}^{+\infty} \left(\sum_{j=0}^{k} b_j \right) \frac{z^k}{k!} = e^{e^z - 1} \tag{18}$$

which can be proven using the property of partial sums for any integer sequence.

4.7. Baseline Methodology

The processing steps of the baseline methodology are extensively described in previous works [23–25]. Initially, for the web documents to be retrieved and later processed, the web document repository (PubMed) is queried. Specifically, we have used PubMed API (Pubmed API: http://www.ncbi.nlm.nih.gov/books/NBK25500/). After the results $D = d_1, d_2, \ldots d_n$ are retrieved in the initial step, each result item d_i consists of six different items: title, author names, abstract, keywords, conference/journal name and publication date, $d_i = \{t_i, an_i, a_i, k_i, cj_i, pd_i\}$.

In the following, the document representation takes place as the proposed methodology enriched the corresponding texts with annotations from a specific ontology. Consequently, each document is represented as a term frequency-inverse document frequency (Tf/Idf) vector, and some terms of the vector are annotated and mapped on senses identified from MeSH.

As a last step, the vectors-documents are clustered by utilizing k-means.

The baseline methodology is outlined in Algorithm 1.

Algorithm 1 Baseline methodology from [25].

Require: Query vector **q**
Ensure: Clusters produced
identify documents set $D = \{d_1, d_2, \ldots, d_n\}$
$\forall$ result item $d_i = \{t_i, an_i, a_i, k_i, cj_i, pd_i\}$ in D
for each d_i in D **do**
    calculation of MeSH vectors $M = \{M_{d_1}, M_{d_2}, \ldots M_{d_n}\}$
end for
use as input, titles, keywords and abstracts: $d_i = \{t_i, k_i\}$
for each M_{d_i} in M **do**
    Tf/Idf Clusters $\leftarrow$ K-Means(M) {where Cosine Similarity metric is applied to k-Means}
end for

5. TREC Dataset and Baseline

5.1. Data Synopsis

In order to compare our proposed tensor based scheme, the TREC Genomics 2007 dataset (TREC Genomics Track: http://ir.ohsu.edu/genomics/) serves as an evaluation benchmark. For a more detailed description of the specific dataset, see [25,50]. It is worthwhile to mention that in the TREC Genomics 2007 dataset, about 160,000 documents from about 50 genomics-related journals are considered.

5.2. Baseline Method

Regarding the clustering procedure, the k-means algorithm is employed with the following parameters. The number of derived clusters is 20, while the cosine similarity distance was utilized for identifying underlying document similarities. Regarding the tensor scheme, Tucker factorization, which is a higher-order SVD generalization, has been executed, and the rows and columns of the base matrices corresponding to the c largest entries of the core tensor have been selected. This is similar to selecting the c largest singular values of the SVD in the matrix case.

We have compared the produced clusters for both schemes by using precision, recall and F-measure scores.

As can be seen in Tables 5–7, the proposed representations and clustering strategies achieve notable precision, recall and F-measure for a small and average number of processed documents. As the number of processed documents increases, the performance of the corresponding methods seems to decrease.

Table 5. Cosine similarity, k-means clustering and precision.

p(%)	tf-idf	tf-idf + MeSH	Tensor	p(%)	tf-idf	tf-idf + MeSH	Tensor
10	77.77	88.98	89.35	60	61.46	59.19	61.93
20	61.38	69.55	69.93	70	48.54	50.93	60.44
30	62.73	61.53	63.38	80	65.49	53.48	58.16
40	63.13	64.74	64.43	90	50.90	61.14	57.96
50	55.79	61.12	63.17	100	50.11	52.27	56.35

Table 6. Cosine similarity, k-means clustering and recall.

p(%)	tf-idf	tf-idf + MeSH	Tensor	p(%)	tf-idf	tf-idf + MeSH	Tensor
10	72.09	83.37	81.93	60	35.27	34.28	33.33
20	48.31	59.44	58.38	70	27.79	32.53	33.11
30	52.15	52.63	50.58	80	30.13	28.19	27.85
40	41.17	51.24	50.23	90	33.67	30.73	27.72
50	30.42	41.18	40.50	100	31.26	30.32	28.56

Table 7. Cosine similarity, k-means clustering and F-measure.

p(%)	tf-idf	tf-idf + MeSH	Tensor	p(%)	tf-idf	tf-idf + MeSH	Tensor
10	74.93	86.99	87.17	60	44.97	45.16	46.75
20	53.74	63.34	64.47	70	37.63	40.14	42.94
30	57.44	54.43	56.64	80	41.11	39.48	42.95
40	50.75	57.12	56.90	90	39.78	42.29	42.63
50	38.36	50.27	52.14	100	38.47	41.13	42.47

By observing Table 8, it is deduced that density is a decreasing function of tensor size. Please notice that $p(\%)$ denotes the percentage of the documents used to extract these results.

Table 8. Tensor density.

p(%)	10	20	30	40	50	60	70	80	90	100
Tensor	72.62	65.81	59.97	53.64	50.12	48.85	46.48	44.63	40.65	34.20

Table 9 shows how $\|\mathcal{G}\|_F$ compares to $\log I_d$. As with density, this ratio falls with I_d. This can be interpreted as the weakening of document connections. When few documents are available, then it is easy to derive strong connections between them. On the other hand, as the collection is augmented with more documents, then topical associations lose in strength due to the increased subject variability.

Table 9. $\|\mathcal{G}\|_F$ ratio to $\log I_d$.

I_d	100	200	500	1.000	2.000	5.000
Ratio	8.85	7.16	5.43	4.68	3.35	2.15

6. Custom PubMed Dataset

Before analyzing the precision and recall characteristics of the proposed model, it is worth looking at the tensor contents and specifically at the term list. The twenty most and the twenty least common keywords in the collection and their frequencies are shown in Table 10. Additionally, Table 11 contains the corresponding information for the text terms. In these tables, the frequency f for both keywords and terms is computed based on the entire document collection.

Table 10. Collection keywords (frequency f as a percentage).

Keyword	f(%)	Keyword	f(%)	Keyword	f(%)	Keyword	f(%)
cognitive	62.18	criteria	16.88	practice	2.45	channel	0.45
cognition	57.91	fMRI	15.41	guide	2.15	handbook	0.42
brain	56.47	HRF	15.36	design	1.67	social	0.36
mind	33.12	EEG	15.22	reliability	1.21	shift	0.22
process	28.11	biosignal	14.83	indicator	1.18	osteoporosis	0.20
thought	27.14	processing	14.21	heart	0.99	shell	0.18
MATLAB	19.23	resolution	14.17	condition	0.78	shock	0.18
methodology	19.17	spatial	13.31	nutrition	0.72	bronchitis	0.17
evaluation	17.75	NIfTI	11.03	unimodal	0.65	open	0.06
clinical	17.02	multimodal	11.02	static	0.58	bradycardia	0.01

It is no surprise to see common technical and medical terms at the top of the list of Table 10. For instance, both fMRI and EEG analysis are widespread techniques with many MATLAB implementations. Moreover, older or more specialized terms are less frequent. For example,

shell shock or shellshock is a rather negatively-charged WWI-era term, which is now largely replaced by post-traumatic. Notice that closely-associated keywords, such as clinical and evaluation, have similar frequencies, which is expected. Rare keywords also pertain to other physiological conditions, probably from papers establishing a connection between brain and body functionality.

Table 11. Collection terms (frequency f as a percentage).

Term	f(%)	Term	f(%)	Term	f(%)	Term	f(%)
cognitive	100.00	condition	65.12	vulnerable	0.04	hydroponics	0.03
cognition	100.00	spatial	61.73	depression	0.04	hazardous	0.03
brain	100.00	temporal	55.24	caregivers	0.04	hypertension	0.02
mind	98.15	vision	47.14	home care	0.04	formulation	0.02
impairment	92.42	age	41.24	portage	0.04	hemospherine	0.02
biosignal	73.34	male	38.24	transfusion	0.04	capacitance	0.02
processing	72.83	female	37.22	jogging	0.04	deleterious	0.02
fMRI	70.11	medication	36.10	abundance	0.04	harbinger	0.01
EEG	68.45	heart	35.34	office	0.04	hypo-robotic	0.01
resolution	67.23	healthy	35.22	Ethiopian	0.03	meta-template	0.01

The situation is similar in Table 11 where the top twenty and the bottom twenty terms are shown. In comparison to Table 10, the terms are more diversified covering a broader number of topics including many secondary ones, and thus, the gaps between terms are considerably narrower. Obviously, the terms cognitive, cognition and brain are present in literally every document of the collection, which was anticipated. In contrast to Table 10, there hardly appears to be a connection between the least frequent terms and the topic. In fact, the right-hand side of Table 10 could appear in virtually any medical collection about any topic and still make some sense. This implies that there is definitely compression potential in the original collection as a portion of documents can be replaced by a combination of eigen-documents or, in the case of redundant information determined by a large number of generic terms, it can be simply discarded.

Notice that the majority of the terms of the second column of Table 11 probably refer to the subjects undergoing some kind of treatment or monitoring. Furthermore, the first column of Table 11 shares many common entries with the corresponding column of Table 10. This can be attributed to the fact that a keyword, which carries significant semantic information, is very likely to be used in the text of a document. Furthermore, the frequency of terms fMRI and EEG equals roughly the sum of the frequency of the academic papers and the clinical data documents of Table 12. A possible explanation is that these types of documents are the most likely to refer to clinical methodology, while the remaining document types address auxiliary topics.

Table 12. Document types (frequency f as a percentage).

Type	f(%)	Type	f(%)
Journal articles	28.12	Best practice handbooks	8.37
Conference papers	26.44	Healthcare reports	5.33
Demographic reports	14.46	Training manuals	3.24
Clinical data	11.02	Other	3.02

Table 12 contains the frequency of each document type in the collection. It comprises approximately half of the scientific papers, which is consistent with the role of PubMed, supplemented by another half of auxiliary documents of various types.

It is of interest to examine the similarity between the keyword set S_k and the term set S_t, as any high relevance between them would mean the tensor can be reduced to a term-document matrix.

Their similarity was assessed with the DTW metric, which works on vectors of lengths p and q. First, it defines a metric between the members of both vectors $\ell_{i,j}$ and then relies on the recurrence relation:

$$\gamma_{i,j} = \ell_{i,j} + \min\left\{\gamma_{i-1,j}, \gamma_{i-1,j-1}, \gamma_{i,j-1}\right\}, \quad 1 \leq i \leq p, 1 \leq j \leq q \tag{19}$$

to compute the shortest transformation and its cost γ^* between those two vectors, creating incrementally a shortest cost path in a $p \times q$ tableau. Since S_k and S_t contain words, $\ell_{i,j}$ was selected to be the Levenshtein distance. One fine point is that DTW requires vectors, which are ordered, whereas sets are by definition unordered. To overcome this, S_k and S_t were sorted in descending order according to word frequency and, if needed, lexicographically, as well, to break any frequency ties. This preserves not only the words in each set, but also their significance. Another subtlety is that the atomic operations are character and not word oriented. Once γ^* was computed, it was expressed as a fraction of the worst case scenario, which is the deletion of each character of S_t followed by the insertion of each character of S_k. For the given sets, this ratio was 0.1181, which means that there is little overlapping between their sorted versions.

Table 13 presents tensor density as a function of I_d. Similarly to the benchmark dataset, it is also a decreasing function of tensor size.

Table 13. Tensor density.

***p*(%)**	10	20	30	40	50	60	70	80	90	100
Tensor	74.57	68.64	62.47	58.88	50.12	46.36	44.92	41.75	37.49	31.16

Table 14 presents the ratio of $\|\mathcal{G}\|_F$ to $\log I_d$. The weakening of document connections is caused similarly to the benchmark dataset case.

Table 14. $\|\mathcal{G}\|_F$ ratio to $\log I_d$.

I_d	100	200	500	1.000	2.000
Ratio	9.13	7.45	5.67	5.02	3.97

7. Conclusions

This article presented a semantically-aware topic-based document clustering scheme for biomedical articles that can be further applied to biomedical ones. The core of this scheme is a keyword-term-document third order tensor, namely a three-dimensional array. The latter is a generalization of the established term-document matrix model, which is widely used in information retrieval, both in research and in industrial-grade systems. A third order keyword-term-document tensor with values coming out a tf-idf scheme is proposed. The advantage of the proposed representation is the semantic enrichment, which is achieved with the inclusion of keywords. Scientometric research suggests that keywords regularly carry more semantic information than ordinary terms. A variation of this model is to mix keywords from MeSH with keywords retrieved from PubMed.

The proposed methodology has been compared to both the term-author-document outlined in [47] in terms of compression potential, precision and recall. Both were implemented in MATLAB using the Tensor Toolbox. The experimental results suggest that inclusion of keywords instead of authors increases precision and, to an extent, recall.

Regarding future work directions, a number of extensions is possible. The sparsity patterns of larger tensors should be analyzed, and if possible, compression techniques such as those proposed in [51] should be applied. Furthermore, effective density patterns should be investigated. Another research point is the addition of update operations to the proposed model, namely of insertion and deletion operations, yielding thus a more flexible scheme. A related topic is the development of

persistent methodologies for tensors, such as those in [52], in order to support the efficient retrieval of past versions. Finally, real-time analytics are gaining attention with the recent combination of streaming algorithms and tensors.

Author Contributions: Georgios Drakopoulos, Andreas Kanavos, Ioannis Karydis, Spyros Sioutas, and Aristidis Vrahatis conceived of the idea, designed and performed the experiments, analyzed the results, drafted the initial manuscript and revised the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Johnson-Laird, P.N. Mental models in cognitive science. *Cognit. Sci.* **1980**, *4*, 71–115.
2. Gover, M.R. The Embodied Mind: Cognitive Science and Human Experience. *Mind Cult. Act.* **1996**, *3*, 295–299.
3. Norman, D.A. Twelve issues for cognitive science. *Cognit. Sci.* **1980**, *4*, 1–32.
4. European Commission. *The 2015 Ageing Report: Underlying Assumptions and Projection Methodologies*; European Commission: Brussels, Belgium, 2014.
5. He, W.; Goodkind, D.; Kowal, P. An ageing world: 2015. In *International Reports*; U.S. Department of Commerce: Washington, DC, USA, 2016.
6. United Nations Population Division. *World Population Ageing 2007*; United Nations Population, New York, NY, USA, 2007.
7. United Nations Population Division. *World Population Ageing 2015*; United Nations Population, New York, NY, USA, 2015.
8. United Nations Population Division. *World Population Prospects 2015*; United Nations Population, New York, NY, USA, 2015.
9. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
10. Newman, M.E. Ego-centered networks and the ripple effect. *Soc. Netw.* **2003**, *25*, 83–95.
11. Newman, D. Writing together separately: Critical discourse and the problems of cross-ethnic co-authorship. *Area* **1996**, *28*, 1–12.
12. Bhattacharya, S.; Ha-Thuc, V.; Srinivasan, P. MeSH: A window into full text for document summarization. *Bioinformatics* **2011**, *27*, 120–128.
13. David, M.R.; Samuel, S. Clustering of PubMed abstracts using nearer terms of the domain. *Bioinformation* **2012**, *8*, 20–25.
14. Theodosiou, T.; Darzentas, N.; Angelis, L.; Ouzounis, C.A. PuReD-MCL: A graph-based PubMed document clustering methodology. *Bioinformatics* **2008**, *24*, 1935–1941.
15. Baud, R.; Rassinoux, A.M.; Scherrer, J.R. Natural language processing and semantical representation of medical texts. *Methods Inf. Med.* **1992**, *31*, 117–125.
16. Alexopoulou, D.; Andreopoulos, B.; Dietze, H.; Doms, A.; Gandon, F.L.; Hakenberg, J.; Khelif, K.; Schroeder, M.; Wächter, T. Biomedical word sense disambiguation with ontologies and metadata: Automation meets accuracy. *BMC Bioinform.* **2009**, *10*, doi:10.1186/1471-2105-10-28.
17. Ananiadou, S.; Thompson, P.; Nawaz, R. Enhancing search: Events and their discourse context. In Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing, Samos, Greece, 24–30 March 2013; pp. 318–334.
18. Mihaila, C.; Ohta, T.; Pyysalo, S.; Ananiadou, S. BioCause: Annotating and analysing causality in the biomedical domain. *BMC Bioinform.* **2013**, *14*, 2.
19. Ontology Consortium, G.E.A. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.* **2004**, *32*, D258–D261.
20. Myhre, S.; Tveit, H.; Mollestad, T.; Lægreid, A. Additional gene ontology structure for improved biological reasoning. *Bioinformatics* **2006**, *22*, 2020–2027.

21. Batista-Navarro, R.T.; Kontonatsios, G.; Mihaila, C.; Thompson, P.; Rak, R.; Nawaz, R.; Korkontzelos, I.; Ananiadou, S. Facilitating the Analysis of Discourse Phenomena in an Interoperable NLP Platform. In Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing), Samos, Greece, 24–30 March 2013; pp. 559–571.
22. Huang, M.; Névéol, A.; Lu, Z. Recommending MeSH terms for annotating biomedical articles. *JAMIA* **2011**, *18*, 660–667.
23. Kanavos, A.; Makris, C.; Theodoridis, E. On Topic Categorization of PubMed Query Results. In Proceedings of the Artificial Intelligence Applications and Innovations (AIAI), Halkidiki, Greece, 27–30 September 2012, pp. 556–565.
24. Kanavos, A.; Theodoridis, E.; Tsakalidis, A. A PubMed Meta Search Engine Based on Biomedical Entity Mining. In Proceedings of the International Workshop on Database and Expert Systems Applications (DEXA), Munich, Germany, 1–5 September 2014; pp. 82–86.
25. Kanavos, A.; Makris, C.; Theodoridis, E. Topic Categorization of Biomedical Abstracts. *Int. J. Artif. Intell. Tools* **2015**, *24*, 1540004.
26. Yoo, I.; Hu, X. Biomedical Ontology MeSH Improves Document Clustering Qualify on MEDLINE Articles: A Comparison Study. In Proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS), Salt Lake City, UT, USA, 22–23 June 2006; pp. 577–582.
27. Boyack, K.W.; Newman, D.; Duhon, R.J.; Klavans, R.; Patek, M.; Biberstine, J.R.; Schijvenaars, B.; Skupin, A.; Ma, N.; Börner, K. Clustering more than two million biomedical publications: Comparing the accuracies of nine text-based similarity approaches. *PLoS ONE* **2011**, *6*, e18029.
28. Yoo, I.; Hu, X. Clustering Large Collection of Biomedical Literature Based on Ontology-Enriched Bipartite Graph Representation and Mutual Refinement Strategy. In Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Singapore, 9–12 April 2006; pp. 303–312.
29. Zhu, S.; Zeng, J.; Mamitsuka, H. Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity. *Bioinformatics* **2009**, *25*, 1944–1951.
30. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500.
31. Kolda, T.G. Orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.* **2001**, *23*, 243–255.
32. Drakopoulos, G. Tensor Fusion of Social Structural and Functional Analytics over Neo4j. In Proceedings of the 6th International Conference of Information, Intelligence, Systems, and Applications, Chandigarh, India, 5–7 March 2016.
33. Drakopoulos, G.; Megalooikonomou, V. A Graph Framework for Multimodal Medical Information Processing. *Proc. Dig. World* **2016**, 1024–1028, arXiv:1608.00134.
34. Drakopoulos, G.; Megalooikonomou, V. An adaptive higher order scheduling policy with an application to biosignal processing. In Proceedings of the 2016 Symposium Series on Computational Intelligence, Athens, Greece, 6–9 December 2016; pp. 921–928.
35. De Lathauwer, L.; De Moor, B.; Vandewalle, J. Independent component analysis based on higher-order statistics only. In Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing, Corfu, Greece, 24–26 June 1996; pp. 356–359.
36. Mourrain, B.; Pan, V.Y. Multivariate polynomials, duality, and structured matrices. *J. Complex.* **2000**, *16*, 110–180.
37. Von zur Gathen, J.; Kaltofen, E. Factoring sparse multivariate polynomials. *J. Comput. Syst. Sci.* **1985**, *31*, 265–287.
38. Kressner, D.; Tobler, C. Algorithm 941: Htucker—A Matlab Toolbox for Tensors in Hierarchical Tucker Format. *ACM Trans. Math. Softw.* **2014**, *40*, 22.
39. Drakopoulos, G.; Megalooikonomou, V. Regularizing large biosignals with finite differences. In Proceedings of the 7th International Conference on Information, Intelligence, Systems, and Applications, Chalkidiki, Greece, 13–15 July 2016; pp. 1–6.
40. Papalexakis, E.E.; Faloutsos, C.; Sidiropoulos, N.D. Tensors for Data Mining and Data Fusion: Models, Applications, and Scalable Algorithms. *TIST* **2016**, *8*, 16.
41. Wang, H.; Zhang, Q.; Yuan, J. Semantically Enhanced Medical Information Retrieval System: A Tensor Factorization Based Approach. Available online: http://ieeexplore.ieee.org/abstract/document/7912400/ (accessed on 17 July 2017).
42. Shadbolt, N.; Berners-Lee, T.; Hall, W. The semantic Web revisited. *IEEE Intell. Syst.* **2006**, *21*, 96–101.

43. Antoniou, G.; Van Harmelen, F. *A Semantic Web Primer*; MIT Press: Cambridge, MA, USA, 2004.
44. Wu, Z.; Palmer, M. Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, Las Cruces, NM, USA, 27–30 June 1994; pp. 133–138.
45. Leacock, C.; Chodorow, M. Combining local context and WordNet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*; The MIT Press: Cambridge, MA, USA, 1998; Volume 49, pp. 265–283.
46. Jiang, J.J.; Conrath, D.W. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv* **1997**, arXiv: preprint cmp-lg/9709008.
47. Drakopoulos, G.; Kanavos, A. Tensor-based Document Retrieval over Neo4j with an Application to PubMed Mining. In Proceedings of the 6th International Conference of Information, Intelligence, Systems, and Applications, Chalkidiki, Greece, 13–15 July 2016.
48. Gómez, L.I.; Gómez, S.A.; Vaisman, A.A. A generic data model and query language for spatiotemporal OLAP cube analysis. In Proceedings of the 15th International Conference on Extending Database Technology, Berlin, Germany, 27–30 March 2012; pp. 300–311.
49. Aggarwal, C.C. *Data Mining: The Textbook*; Springer: Berlin, Germany, 2015.
50. Hersh, W.R.; Cohen, A.M.; Ruslen, L.; Roberts, P.M. TREC 2007 Genomics Track Overview. In Proceedings of The Sixteenth Text REtrieval Conference, (TREC), Gaithersburg, MD, USA, 6–9 November 2007.
51. Drakopoulos, G.; Megalooikonomou, V. On the weight sparsity of multilayer preceptrons. In Proceedings of the 6th International Conference on Information, Intelligence, Systems, and Applications, Corfu, Greece, 6–8 July 2015; pp. 1–6.
52. Kontopoulos, S.; Drakopoulos, G. A space efficient scheme for graph representation. In Proceedings of the 26th International Conference on Tools with Artificial Intelligence (ICTAI 2014), Limassol, Cyprus, 10–12 November 2014; pp. 299–303.