Major Project Thesis

On

CNESSI: Citation Network with Enhanced Structural and Semantic Information

Submitted by

Bhuvan M S

12IT16

Under the Guidance of,

Prof. Ananthanarayana V S

Department of Information Technology, NITK, Surathkal

Prof. M Narasimha Murty

Mr. Govind Sharma

Department of Computer Science and Automation, IISc, Bangalore

Date of Submission: 23 Nov, 2015

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

2015-2016

Department of Information Technology, NITK Surathkal
Major Project I & II
B. Tech Thesis (2015 - 2016)

Course Code : IT 449 & IT 499

Course Title: Major Project I & II

Project Title: *CNESSI: Citation Network with Enhanced Structural and Semantic Information*

Project Group:

Name of the Student	Register No.	Signature with Date
Bhuvan M S	12IT16 (122353)	

Place: NITK, Surathkal

Date: 23 Nov, 2015

(Prof. Ananthanarayana V S)

Place: IISc, Bangalore

Date: 27 Nov, 2015

(Prof. M Narasimha Murty)

Acknowledgements

My sincere gratitude to department of Information Technology at National Institute of Technology Karnataka, Surathkal, for permitting to carry out the major project at Indian Institute of Science, Bangalore. I would like to thank Prof. M. Narasimha Murty and Prof. Ananthanarayana. V. S for guiding me along the path of my research. Great thanks to Mr. Govind Sharma who has always given me expert suggestions during the research. Thanks to people who opened up the ACL ARC dataset and respective metadata enabling this research.

Abstract

Citation network analysis helps to extract and identify the characteristics of scientific research and help in better indexing and organization of digital libraries and ranking of research documents and authors based on their impact measure through citations. Current state of analysis and indexing applications including Google Scholar are based on the syntactic information like the citation count, rather than the semantic information like the amount of knowledge represented in a research document. The factor limiting to perform such superior semantic analysis is the lack of structural and semantic information in the citation network. The research presents a novel methodology to create a Citation Network with Enhanced Structural and Semantic Information (CNESSI) from raw citation networks starting from raw PDF files of research documents. This has been achieved through a hybrid methodology of six phases, each solving a significant structural or semantic problem. Initial phases achieve structural information extraction by intelligent tagging of structural parts of the research documents using Machine Learning algorithms like Support Vector Machines. The latter sections achieve the semantic information extraction and tagging with respect to knowledge domains using Natural Language Processing and Topic Synthesis techniques like Latent Dirichlet Allocation. The discussion and analysis of each phase of the hybrid methodology and the kind of observation that lead to decisions which shaped the methodology have been comprehensively presented. Further, the usage of added structural and semantic information in citation network and the research options expanded by CNESSI, enabling superior semantic analysis of citation network have been discussed. The interesting results that could be obtained through semantic analysis on CNESSI could reveal profusion of insights of paramount significance.

Keywords: *Natural Language Processing; Machine Learning; Support Vector Machines; Latent Dirichlet Allocation; Topic Analysis; Scientometry; Citation Network; Information Extraction*

Contents

List of Figures

List of Tables

Abbreviations

- KT: Knowledge Topic referring to a particular Knowledge Domain.
- Different KTs: 'NLP - Natural Language Processing' (KT1), 'IR - Information Retrieval' - (KT2), 'ML - Machine Learning' (KT3), 'DAA - Data structures and Algorithms' (KT4), 'HPC - Systems and High Performance Computing' (KT5).
- Raw-Un-Validated-Sections: Raw sections specific to each research document, extracted from PDF which are not validated.
- Raw-section: Valid Raw sections specific to each research document, which are validated using rule based section validation in phase 2.
- Section: Generic Section, Generic Part, Normalized Section, is a Valid Raw section normalized into one of generic sections by classifying after phase 3. Each text in document is organized into the below sections from phase 3 onwards.
- Different Sections: 'Motivation' (S1), 'Background' (S2), 'Literature Review' (S3), 'Methodology' (S4), 'Results and Discussion' (S5) and 'Conclusion and Future Work' (S6).
- Text-chunk: Text of each section and each citation context. All text-chunks refers to all the sections and all the citation contexts of all the research documents. Text-chunks of sections means all the texts of all sections of all documents. Text-chunks of citations means all the texts of all citation contexts of all documents.
- T: Topics, as extracted from LDA.
- S: Section.
- D: All text-chunks.
- W: Set of words/terms representing all the text-chunks.
- W': Set of words/terms representing all the Section at the corpus level.
- W": Set of words/terms representing all the KT.
- gi: index for i-th section in the matrix at the corpus level.

- si: index for i-th section in the matrix present the node (research document).

- KSV: Knowledge-Share-Vector; is a vector of dimension 5, where each dimension represents a KT. The values in these vectors are always less than 1 as they are a sub portion of probabilistic distribution of KT against terms (W").

- A-B matrix: It is a matrix with set A along the rows and set B along the columns header as shown in below matrx.

$$\begin{array}{cc}
\underline{\textbf{A-B}} & \begin{array}{cccc} a1 & a2 & \dots & an \end{array} \\
\begin{array}{c} b1 \\ b2 \\ \vdots \\ bn \end{array} & \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots \end{pmatrix}
\end{array}$$

- df: Document Frequency used for representing any text.

- tf: Term Frequency used for representing any text.

- cite-code: The placeholder at the point of citation in the citing document, where it refers to the cited document (like [1], or (last-name, year) etc.).

- fig-code: Figure reference in text (like, as shown in Figure 1).

1 Introduction

1.1 *Motivation*

Scholars across the world contribute to science through research documents conference papers, journals, articles etc. These articles represent novel work in different field. It is very important to follow the development and evaluate contributions done by each of these research documents. Citations present in each of these documents bibliography section gives us a way to understand where what part of the document was inspired by which other document. This ultimately forms a network of citations. More formally it would form a network of knowledge flow which when built can be used to solve infinite new analysis problems which could help us derive insights into how different domains of knowledge are developing through time. Also it would help us evaluate these research documents to derive how much inspiration each of them provide to the scientific community through time.

An author after publishing his paper should be able to get feedback regarding how much knowledge share existed in his research work, how his research work has impacted other research documents with respect to different knowledge domains. Ranking authors and research documents overall could be very ambiguous to interpret, perhaps a finer ranking with respect to different knowledge domains could be very useful. More enthusiastic results must be possible to derive from citation networks like, determining the knowledge domain any set of authors generally work on, and which knowledge domain had more impact on which knowledge domain. Also, along the time line, the author must be able to track the shift in the knowledge domain in the author's research documents.

All these problems are highly interesting and could have significant impact to improve the standard of knowledge and research tracking among the scientific community. It could also accelerate the scientific development due to clear and organized feedback establishment through knowledge domain based evaluation of research documents and authors. Perhaps, it could help any research community decide how much they should focus on different knowledge domains. It can help a new research find all resources organized according to knowledge domain.

But till today, such analysis and such insights are not found. Even Google Scholar

[32] ranks the research documents based on the citation counts. Google scholar does not have the information on how much knowledge a particular research document has represented. Each citation is not of equal importance [43]. Perhaps, people would cite an famous document just for the sake that is famous document and need to be cited, even without reading it. Such citations must be given less importance, by verifying the knowledge share in the two documents. Other than ranking, all problems like community detection [9], sleeping beauty identification [23] etc.. are done based on citation counts, and not the knowledge represented by the research document.

Fine grained analysis would demand to understand inspiration per domain of knowledge has been provided by a research document. This raises the need to measure the knowledge represented in text of these research documents in each of these sections, and in turn in the text of the context of citations in the research documents. Such knowledge quantization would enable us to build more informative citation network. Social network and graph problems when applied on this citation network would give us much more information. For instance, a standard sleeping beauty problem (finding the research document, which grows very influential not until another work gets invented later in time), would now give fine grained information like which part of that research document was the sleeping beauty. And also help us answer questions like what domain of knowledge represented was actually wake up after the new innovation. Clearly such information is very valuable in the research field.

More concentration just on count of citations. Current Citation Networks are a heterogeneous social network built in mere syntactic representation, but we require a highly structural and semantic information rich citation network as shown in Figure 1.1. Such citation network would enable interesting research as discussed above, hence improve the way research documents and authors are ranked and in fact determine how the knowledge is flowing the research community. Digital Libraries built and indexed using such enhanced citation network could evolve to be more effective as it is more convenient for researches due to superior organization and semantic representation of research documents.

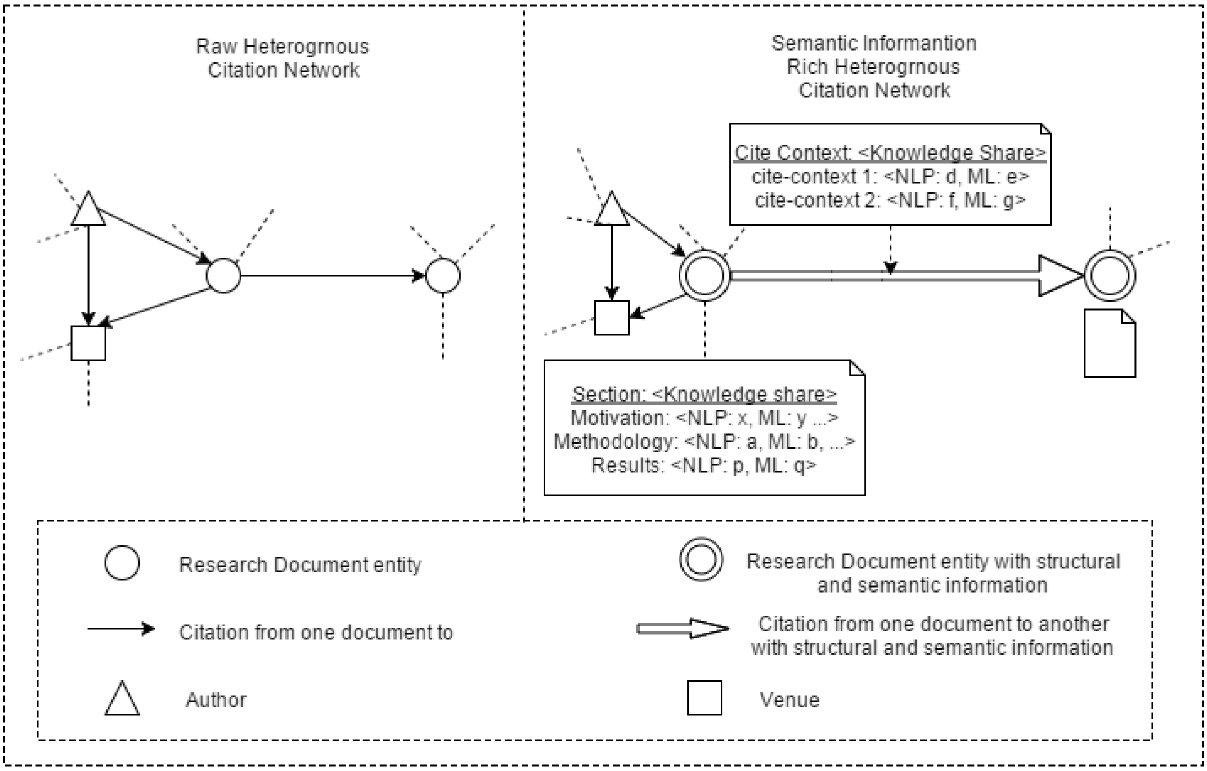

Raw Heterogrnous
Citation Network
Semantic Informantion
Rich Heterogrnous
Citation Network
Cite Context: <Knowledge Share>
cite-context 1: <NLP: d, ML: e>
cite-context 2: <NLP: f, ML: g>
Section: <Knowledge share>
Motivation: <NLP: x, ML: y ...>
Methodology: <NLP: a, ML: b, ...>
Results: <NLP: p, ML: q>
Research Document entity
Research Document entity with structural and semantic information
Citation from one document to
Citation from one document to another with structural and semantic information
Author
Venue

Figure 1.1: Comparing a Plain Citation Network with a Semantically enhanced Citation network.

1.2 *Background*

This chapter describes some concepts that are necessary as a prerequisite to understand the subsequent chapters.

1.2.1 *PDF format*

The research documents are found generally in the PDF format. It serves the human readable purposes mainly. Portable Document Format (PDF) [37] is a file format used to present documents in a manner independent of application software, hardware, and operating systems. Each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, graphics and other information needed to display it. PDF files can contain two types of metadata. The first is the Document Information Dictionary, a set of key-value fields such as author, title, subject, creation and update dates. This is stored in the optional Info trailer of the file. A small set of fields is defined, and can be extended with additional text values if required.

PDF may be generated through multiple ways, from Microsoft word documents, text file, raw images, but the most consistent PDF file if created through Latex [34]. Hence it is not deterministically possible to convert any PDF back to Latex or any machine readable structured format. Hence hence extraction and the document structure extraction from PDF is not very successful. If the PDF has been created from images, then it is not possible to get the text information clearly since the image processing to extract text from images is not trivial. The technical issue occurs when PDF files are created from scanned documents. PDF files created by scanning hard-copy documents containing primarily text do not have the same structure as a PDF file of the same document created directly. The scanned document internally contains a picture of the document, with no information about the text. As far as a user can see it is just another PDF file, with a name and extension indistinguishable from any other; a good scan may look exactly the same as a native PDF file, although a visually poor-quality file, often with skewed pages, gives away its nature. However, the file size will be different, and it will not be possible to search for text. For a scan of adequate quality it is possible with suitable software to regenerate the text of the document with Optical character recognition (OCR), and embed it in the file so as to make it search-able, subject to the accuracy of the OCR [37].

1.2.2 *Citation Network*

Citation Network is a Social network which contains paper sources and linked by co-citation relationships. Egghe & Rousseau [8] explain when a document di cites a document dj, we can show this by an arrow going from the node representing di to the document representing dj. In this way the documents from a collection D form a directed graph, which is called a citation graph or citation network.

Citation is a reference to a published or unpublished source (not always the original source). More precisely, a citation is an abbreviated alphanumeric expression embedded in the body of an intellectual work that denotes an entry in the bibliographic references section of the work for the purpose of acknowledging the relevance of the works of others to the topic of discussion at the spot where the citation appears. Context of a citation is the context in which the citing document has cited the cited document.

The basic structure is mentioned below [13]

- Citation networks are directed. The links go from one document to the other.

- Citation networks are acyclic because a paper can cite only existing papers.

- All edges in the citation networks point backwards in time.

- Links in co-authorship networks are reciprocal (symmetric).

- The link weights between two authors in co-authorship networks can increase over time if they have further collaboration.

- Vertices and edges added to the citation/co-authorship networks are permanent and cannot be removed at a later time.

- The already formed part of the network is mostly static; only the leading edge of the network changes.

1.2.3 *LDA (Latent Dirichlet Allocation) for Topics Analysis*

In natural language processing, Latent Dirichlet allocation (LDA) [33] is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA is an example of a topic model and was first presented as a graphical model for topic discovery.

In LDA, each document may be viewed as a mixture of various topics. This is similar to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a Dirichlet prior. In practice, this results in more reasonable mixtures of topics in a document. It has been noted, however, that the pLSA model is equivalent to the LDA model under a uniform Dirichlet prior distribution [10].

For an intuitive explanation; Suppose you have the following set of sentences:

- I **eat fish** and **vegetables**.

- *Fish* are *pets*.

- My *kitten* **eats fish**.

LDA is a technique that automatically discovers topics that these documents contain. Given the above sentences, LDA might classify the bold words under the Topic F, which we might label as food. Similarly, words in italics might be classified under a separate Topic P, which we might label as pets. LDA defines each topic as a bag of words, and you have to label the topics as you deem fit [33]. Hence, it is an unsupervised technique of soft clustering.

It performs matrix factorization. For a given Document-Term matrix, it factorizes it into Document-Topics (Each Document represented as a probability distribution across Topics for the number of topics given to LDA) and Topic-Word matrix (Each Topic represented as a probability distribution across words in the vocabulary). Detailed information with mathematical background can be found in [4]

1.2.4 *SVM (Support Vector Machines) for Classification*

In machine learning, support vector machines (SVMs, also support vector networks [11]) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [41].

Multi-class classification can be done using One-versus-One scheme [3]. In the one-vs.-one (OvO) reduction for K classes, one trains K (K 1) / 2 binary classifiers for a K-way multi-class problem; each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all K (K 1) / 2 classifiers are applied to an unseen sample and the class that got the highest number of ”+1” predictions gets predicted by the combined classifier.

The parameter is the penalty parameter for the error term. It tells the SVM optimization how much to avoid miss-classifying each training sample. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane miss-classifies more points. For very tiny values of C, you should get misclassified examples, often even if your training data is linearly separable [24]. Hence, depending on the dataset, this parameter need to be tuned using cross-validation.

1.2.5 ***Auxiliary concepts***

Some auxiliary concepts that are used to manipulate data are described briefly.

- *Cosine Similarity:* Cosine similarity [29] is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0 is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90 have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

- *Stop Words:* In computing, stop words [40] are words which are filtered out before or after processing of natural language data (text). Though stop words usually refer to the most common words in a language, there is no single universal list of stop words used by all processing of natural language tools, and indeed not all tools even use such a list. Some tools specifically avoid removing these stop words to support phrase search.

- *Porter Stemmer:* Stemming [39] is the term used in linguistic morphology and information retrieval to describe the process for reducing inflected (or sometimes derived) words to their word stem, base or root formgenerally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root.

  The Porter stemming algorithm [21] (or 'Porter stemmer') is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.

- *Levenshtein Distance:* In information theory and computer science, the Levenshtein distance [35] is a string metric for measuring the difference between two sequences.

Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. Levenshtein distance may also be referred to as edit distance.

- *Bag of Words:* In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features [27].

- *Parts-of-Speech (POS) Tagger:* In corpus linguistics, part-of-speech tagging [36] (POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its contexti.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc.

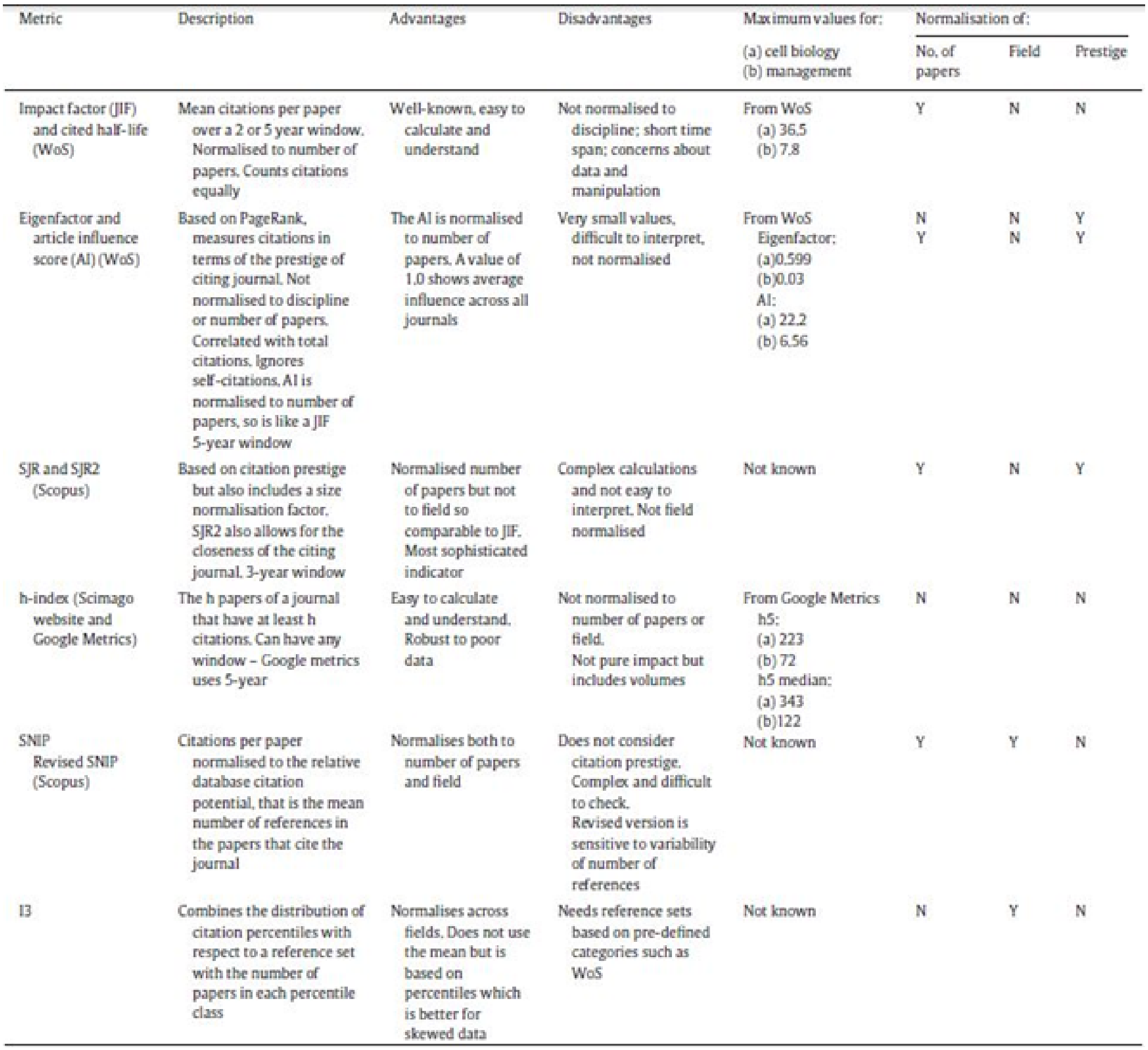
2 Literature Review

This section reviews the related works done previously. Initially, the state-of-the-art methods for text extraction from PDF documents in a partly structured format are reviewed which could be used for the initial phase of this research. Subsequently, a review of past analysis done based on citation networks is provided. Finally, the reasons limiting the analysis on citation networks are observed which stand as basis for formal problem specification in order to enhance the citation network such that it would overcome these limitations and enable semantically rich citation analysis.

2.1 *PDF to computer readable format*

PDF are not structured enough to be deterministically read by machines. Hence, they need to be converted to more reliable and consistent format. GROBID [14] (or Grobid; means GeneRation Of BIbliographic Data) is a machine learning library for extracting, parsing and re-structuring raw documents such as PDF into structured TEI-encoded documents with a particular focus on technical and scientific publications.

GROBID also includes a state of the art model for the extraction and the recognition of bibliographic citations. The references present in an article or patent are identified, parsed, normalized and can be matched with a standard reference database such as Cross-Ref or DocDB (patents). Citation information is considered very useful for improving search ranking and makes it possible to run bibliographic studies and graph-based social analyses. For instance, the citation notification service of ResearchGate uses GROBID bibliographic reference extraction to process every uploaded article. When an existing publication of a registered member is identified, the member can be informed where and how his work has been cited. More challenging, the restructuring of the body of a document (potentially including figures, formula, tables, footnotes, etc.) is continually improving and is currently the object of the semi-automatic generation of more training data. Although more experimental, it can provide to a search engine for scientific literature richer and better text content and structures than basic PDF extractors (e.g., pdftotext, Apache TIKA or PDFBox) [15]. Since the document body restructuring and organisation into sections and respective paragraphs GROBID is chosen to convert from PDF to XML.



Metric	Description	Advantages	Disadvantages	Maximum values for:	Normalisation of:		
				(a) cell biology (b) management	No. of papers	Field	Prestige
Impact factor (JIF) and cited half-life (WoS)	Mean citations per paper over a 2 or 5 year window. Normalised to number of papers. Counts citations equally	Well-known, easy to calculate and understand	Not normalised to discipline; short time span; concerns about data and manipulation	From WoS (a) 36.5 (b) 7.8	Y	N	N
Eigenfactor and article influence score (AI) (WoS)	Based on PageRank, measures citations in terms of the prestige of citing journal. Not normalised to discipline or number of papers. Correlated with total citations. Ignores self-citations. AI is normalised to number of papers, so is like a JIF 5-year window	The AI is normalised to number of papers. A value of 1.0 shows average influence across all journals	Very small values, difficult to interpret, not normalised	From WoS Eigenfactor: (a)0.599 (b)0.03 AI: (a) 22.2 (b) 6.56	N Y	N N	Y Y
SJR and SJR2 (Scopus)	Based on citation prestige but also includes a size normalisation factor. SJR2 also allows for the closeness of the citing journal. 3-year window	Normalised number of papers but not to field so comparable to JIF. Most sophisticated indicator	Complex calculations and not easy to interpret. Not field normalised	Not known	Y	N	Y
h-index (Scimago website and Google Metrics)	The h papers of a journal that have at least h citations. Can have any window – Google metrics uses 5-year	Easy to calculate and understand. Robust to poor data	Not normalised to number of papers or field. Not pure impact but includes volumes	From Google Metrics h5: (a) 223 (b) 72 h5 median: (a) 343 (b)122	N	N	N
SNIP Revised SNIP (Scopus)	Citations per paper normalised to the relative database citation potential, that is the mean number of references in the papers that cite the journal	Normalises both to number of papers and field	Does not consider citation prestige. Complex and difficult to check. Revised version is sensitive to variability of number of references	Not known	Y	Y	N
I3	Combines the distribution of citation percentiles with respect to a reference set with the number of papers in each percentile class	Normalises across fields. Does not use the mean but is based on percentiles which is better for skewed data	Needs reference sets based on pre-defined categories such as WoS	Not known	N	Y	N

Figure 2.1: Comparison of previous work from Review of Scientometrics [16].

2.2 *Review of Scientometry*

Scientometrics was first defined by Nalimov (1971,p.2) as developing the quantitative methods of the research on the development of science as an informational process. It can be considered as the study of the quantitative aspects of science and technology seen as a process of communication. Some of the main themes include ways of measuring research quality and impact, understanding the processes of citations, mapping scientific fields and the use of indicators in research policy and management. Scientometrics focuses on communication in the sciences, the social sciences, and the humanities among several related fields [16]:

- *Bibliometrics* The application of mathematics and statistical methods to books and other media of communication (Pritchard,1969,p.349). This is the original area of study covering books and publications generally. The term bibliometrics was first proposed by Otlet(1934); see also Rousseau(2014).

- *Informetrics* The study of the application of mathematical methods to the objects of information science (Nacke,1979,p.220). Per-haps the most general field covering all types of information regardless of former origin (BarIlan,2008; Egghe&Rousseau, 1990; Egghe&Rousseau, 1988; Wilson,1999).

- *Webometrics* The study of the quantitative aspects of the construction and use of information resources, structures and technologies on the Web drawing on bibliometric and informetric approaches (Bjrneborn&Ingwersen ,2004,p.1217; Thelwall&Vaughan, 2004; Thelwall,Vaughan,&Bjrneborn, 2005). This field mainly concerns the analysis of web pages as if they were documents.

- *Altmetrics* The study and use of scholarly impact measures based on activity in online tools and environments (Priem,2014,p.266). Also called Scientometrics2.0, this field replaces journal citations with impacts in social networking tools such as views, downloads, likes, blogs, Twitter, Mendelay, CiteULike.

2.3 *Previous Attempts on Citation Network Enhancements*

Below are some applications with significant enhancements to citation network and indexing of scholarly articles.

1. *Google Scholar:* Google Scholar [32] is a freely accessible web search engine that indexes the full text or metadata of scholarly literature across an array of publishing formats and disciplines.

2. *Scopus:* Scopus [38] is a bibliographic database containing abstracts and citations for academic journal articles. It covers nearly 22,000 titles from over 5,000 publishers, of which 20,000 are peer-reviewed journals in the scientific, technical, medical, and social sciences (including arts and humanities).

3. *Microsoft Academic Graph (MAG):* The Microsoft Academic Graph [20] is a heterogeneous graph containing scientific publication records, citation relationships

between those publications, as well as authors, institutions, journals and conference "venues" and fields of study. But the fields of study are mapped to keywords, which in turn are mapped to papers. But the semantic information in the text of these papers are not used to determine the fields of study.

4. *CiteSeer:* CiteSeer [28] was a public search engine and digital library for scientific and academic papers, primarily in the fields of computer and information science that has been replaced by CiteSeerX.

    - Autonomous Citation Indexing automatically created a citation index that can be used for literature search and evaluation.
    - Citation statistics and related documents were computed for all articles cited in the database, not just the indexed articles.
    - Reference linking allowing browsing of the database using citation links.
    - Citation context showed the context of citations to a given paper, allowing a researcher to quickly and easily see what other researchers have to say about an article of interest.
    - Related documents were shown using citation and word based measures and an active and continuously updated bibliography is shown for each document.

    It is often considered to be the first automated citation indexing system, has a patent on this topic, and was considered a predecessor of academic search tools such as Google Scholar and Microsoft Academic Search.

5. *ArnetMiner:* Arnetminer [25] is designed to search and perform data mining operations against academic publications on the Internet, using social network analysis to identify connections between researchers, conferences, and publications. This allows it to provide services such as expert finding, geographic search, reviewer recommendation, association search, course search, academic performance evaluation. Arnetminer is commonly used in academia to identify relationships between and draw statistical correlations about research and researchers.

2.4 *Outcome of Literature Survey*

Since can provide to a search engine for scientific literature richer and better text content and structures than basic PDF extractors (e.g., pdftotext, Apache TIKA or PDFBox) [15]. Since the document body restructuring and organisation into sections and respective paragraphs GROBID is chosen to convert from PDF to XML.

The metrics of scientometry are centered around citation counts like total number of citations, number of citations per paper of a journal and so on. Normalizations were done on these citation counts to come up with more informative metrics like Crown indicator measure for evaluating research units suitably normalised against other factors (Waltman, vanEck, vanLeeuwen, Visser, &vanRaan, 2010, 2011). The Figure 2.1 provides a comparative view into the past literature in Scientometry. As we can observe, ranking and similar evaluating study has been done considering citation counts but work has not been done where research documents are ranked based on the knowledge intersection with the citing documents. The latter is clearly more semantic information based study which needs Citation Network with enhanced structural and semantic information. TO the best of knowledge, no research has been done to extract the knowledge represented by each section of each research document, thereby structurally and semantically enhancing the Citation Network.

From all the previous works on enhancements of citation networks and indexing, clearly one can observe that there has been concentration on syntactic information extraction like the author and research paper and their affiliations. MAG has extracted fields of study with respect to each paper, but the fields of study are mapped to keywords, which in turn are mapped to papers. But the semantic information in the text of these papers are not used to determine the fields of study. Hence, the semantic information extraction needs to be done from the text of the research documents to semantically enhance the citation network. Ranking and indexing across more documents is facilitated from all the above methods. But they still do not contain the information needed to perform analysis as mentioned in the Motivation chapter.

The previous works have concentrated towards more syntactic enhancements to the citation networks. This res each concentrates on structural and semantic enhancement as shown in Figure 2.2. Though enhances the Citation Network, it does not open up more research options to perform semantic analysis to mine interesting insights. Thereby,

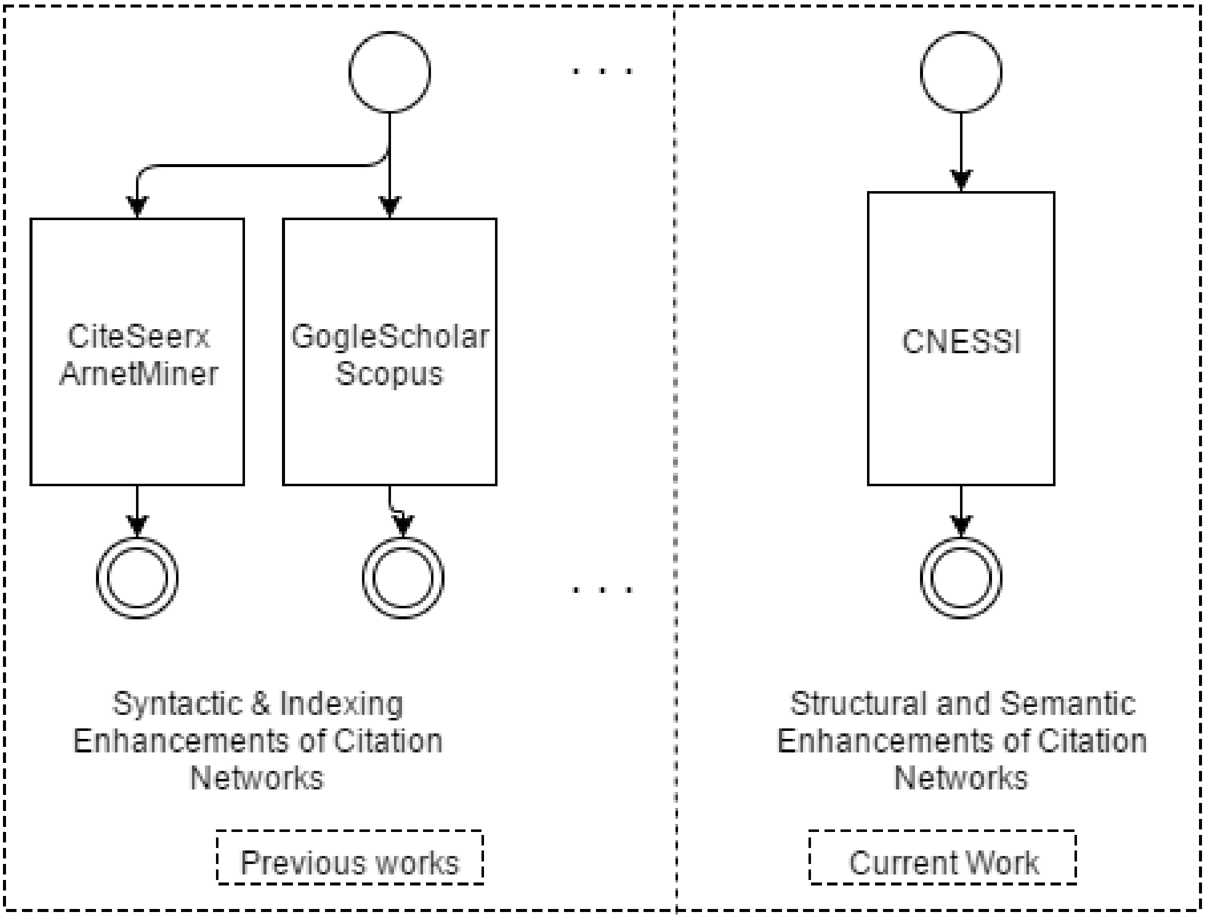

. . .
CiteSeerx
ArnetMiner
GogleScholar
Scopus
. . .
CNESSI
Syntactic & Indexing Enhancements of Citation Networks
Previous works
Structural and Semantic Enhancements of Citation Networks
Current Work

Figure 2.2: Depicting the different analysis needed to Enhance plain citation networks.

the bottom-line is that unless things develop structurally and semantically mapping it to knowledge domain at the source level of citation network, the horizon of citation analysis cannot be widened.

2.5 *Problem Statement*

The problem statement is so chosen that is adds structural and semantic information to the citation network itself, such that the enhanced citation network would open up numerous research options as mentioned in the Motivation Chapter. The problem statement is to take the raw Citation Network as input with the PDF for each research document representing the node. Construct an Information Extraction model to add Structural (normalised section-wise organization) and Semantic (mapping each section and citation context to knowledge domain by analysing the text pertaining to the section) information to the get enhanced Citation Network with additional details on each node and edge as shown in Figure 2.3. This creates CNESSI: Citation Network with Enhanced Structural and Semantic Information.

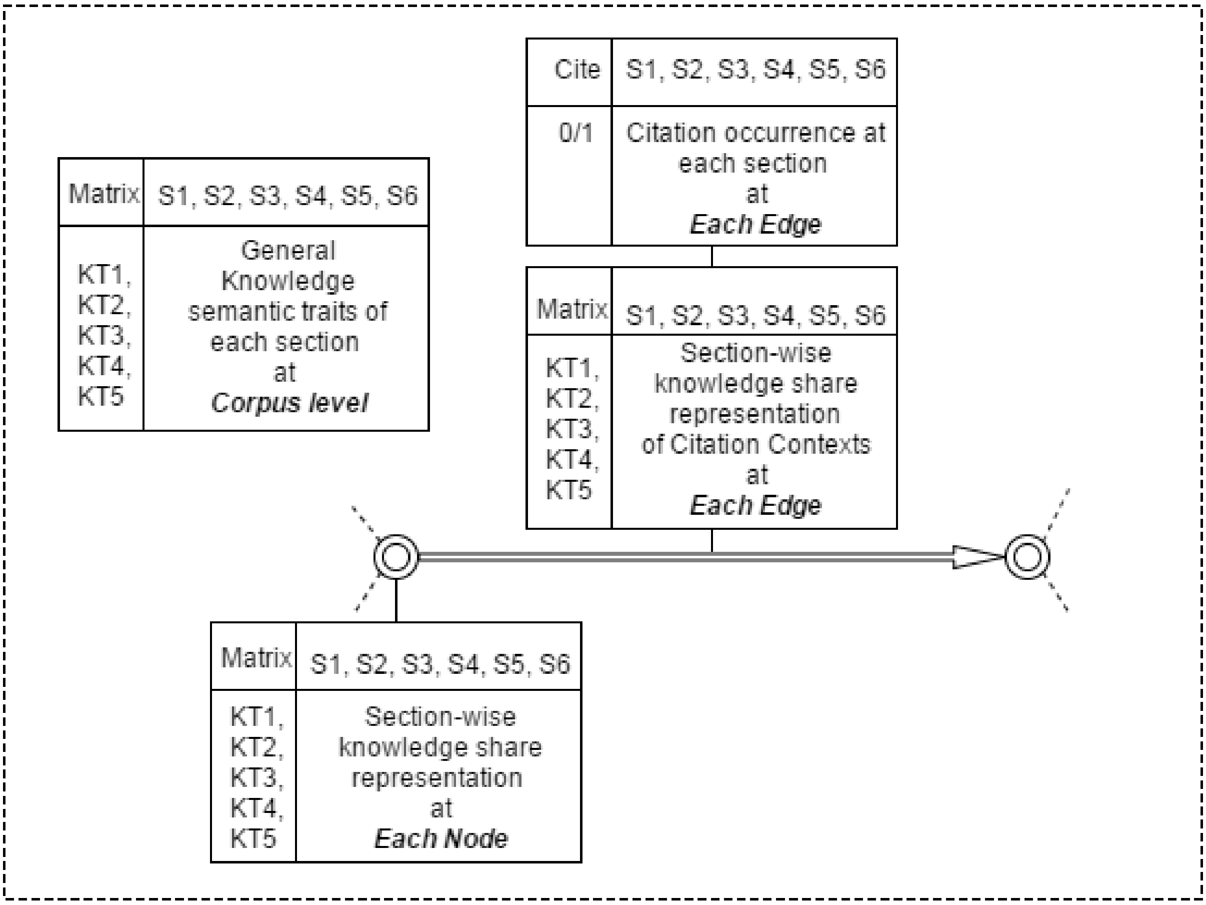
Note that the columns with respect each section in the node information shown in Figure 2.3 are the KSV pertaining to the section in the document representing the node. The columns with respect each section in the edge information shown in Figure 2.3 are the KSV pertaining to the Citation Context in this section of citing paper. The matrix on the edge shown at the top is is 0 or 1 matrix, which specified if this citation has taken place from the respective section or not (since a citation to a document can be done from multiples sections). Similarly in the corpus level matrix, columns with respect to each section are KSV of the section across the corpus in general. (Note that the values in KSV are always less than 1 as they are a sub portion of probabilistic distribution of KT against terms (W”))

The above defined problem can be broken down into following set of objectives mentioned in the below subsection.

2.6 *Objectives*

copy with little alteration the objectives written previously.

1. Obtaining standard dataset of research documents as pdf and high impact publish-



Matrix	S1, S2, S3, S4, S5, S6
KT1, KT2, KT3, KT4, KT5	General Knowledge semantic traits of each section at ***Corpus level***

Cite	S1, S2, S3, S4, S5, S6
0/1	Citation occurrence at each section at ***Each Edge***

Matrix	S1, S2, S3, S4, S5, S6
KT1, KT2, KT3, KT4, KT5	Section-wise knowledge share representation of Citation Contexts at ***Each Edge***

Matrix	S1, S2, S3, S4, S5, S6
KT1, KT2, KT3, KT4, KT5	Section-wise knowledge share representation at ***Each Node***

Figure 2.3: View of CNESSI, showing the structural and semantic information on each node and edge of Citation Network.

ers.

2. Constructing a structured representation of research documents with section-wise organization where each section text is organized to contain paragraph texts and figures information along with the citation done from the section.

3. Validating the raw-section acquired after completion of above objective, and loss less merging of raw-section information into valid set of raw-section.

4. Classifying the raw-sections into generic-parts or 'normalized-sections' (Motivation, Literature Survey, Methodology, Results, Conclusion and Future Work) which exist in every document. (From this phase onwards referred to as simply sections)

5. Identifying the Knowledge Topics (NLP (Natural Language Processing), IR (Information Retrieval), ML (Machine Learning), DAA (Data structures And Algorithms), HPC (Systems and High Performance Computing)), in turn mine the terms which represent this Knowledge Topic.

6. Extract generic traits of each section and find KSV for each section at the corpus level.

7. Computing the Knowledge-Share-Vector (KSV) of each section and citation context, which can be realized as the information on each node and each edge respectively.

3 Methodology

The chapter is organized by first describing the overview of the methodology where it describes how the problem is broken down into phases and how different phases are integrated into the model to solve the greater goal. In each subsection of the chapter, each phase is didactically described. At each subsection, the general methodology and approach of the phase is described followed by detailed implementation details (Python 2.7.6 has been used as the coding language for implementing all the phases). Finally, it is described how to realize the final output as a Citation Networked with Enhanced Structural Semantic Information.

3.1 *Overview of Methodology*

The approach to solve the problem can be broken down into 6 different phases, from the inputting the raw PDF research documents to building structural semantic information rich citation network. Figure 3.1 represents how different phases fit together in the complete design.

The dataset or the corpus contains 10951 ACL research documents and metadata with respect to each research document. But note that the same methodology can be applied to any dataset (which contains metadata; or after extracting metadata). These are in PDF format which cannot be structurally read by machines. Hence, it needs to be converted to more consistent readable format, like Python Objects [18]. The first phase takes this responsibility where each PDF document is first converted to XML [42] and then into Python Objects. It also gives IDs to each research document and authors for unique identification. These IDs are identifiers for each research document aiding it to be represented as node in citation network. Also, it extracts various sections of the research documents and their respective paragraphs and figure captions and descriptions. These sections extracted directly from PDF are not validated here hence are referred as 'Raw Un-Validated Sections'.

The phase 2, is responsible for validating each raw-unvalidated section using a custom rule based validating algorithm, where invalid sections are merged into previous sections after analysing their reasons for invalidity. This phase outputs documents with raw-sections.

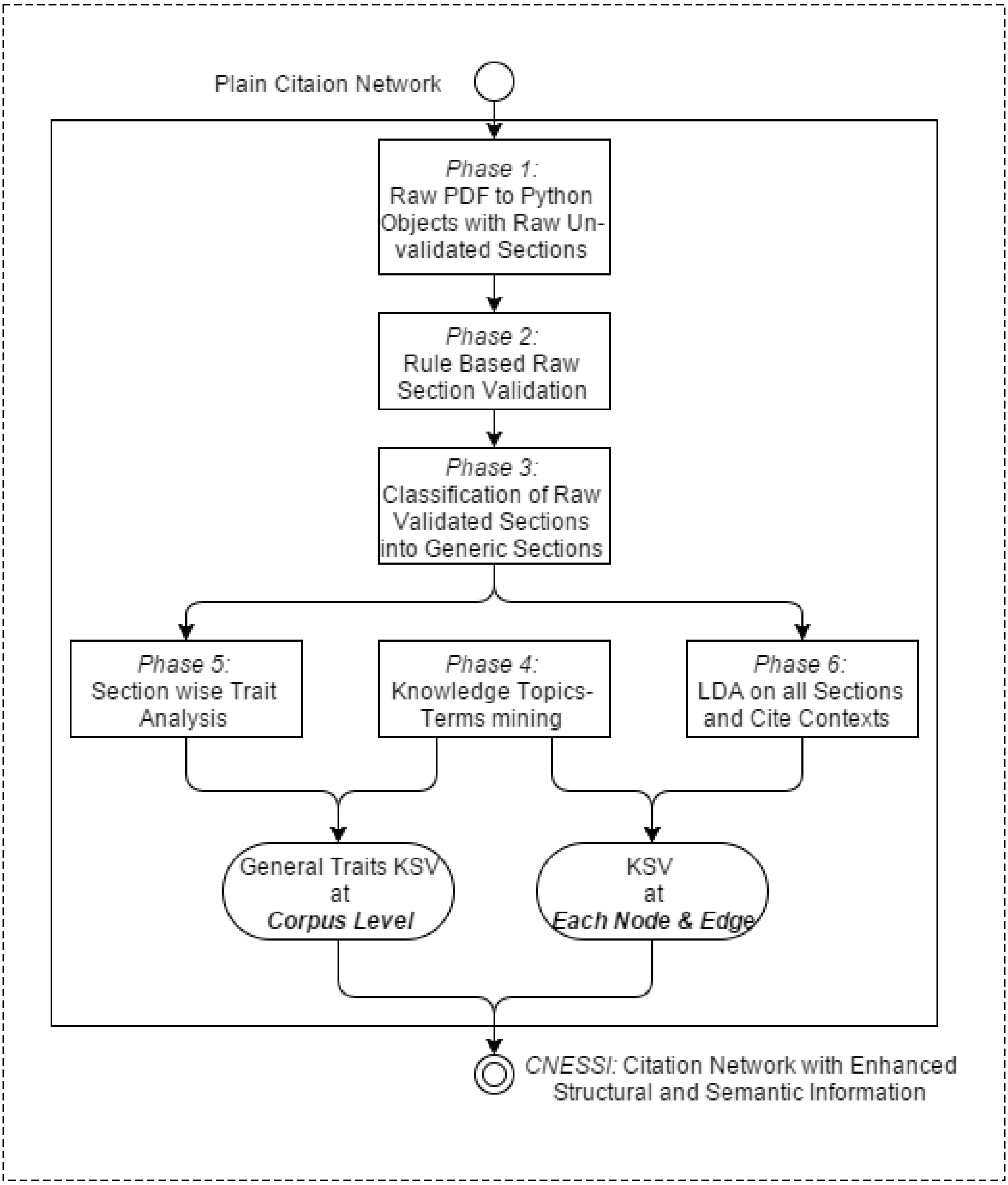

Plain Citaion Network
Phase 1: Raw PDF to Python Objects with Raw Un-validated Sections
Phase 2: Rule Based Raw Section Validation
Phase 3: Classification of Raw Validated Sections into Generic Sections
Phase 5: Section wise Trait Analysis
Phase 4: Knowledge Topics-Terms mining
Phase 6: LDA on all Sections and Cite Contexts
General Traits KSV at Corpus Level
KSV at Each Node & Edge
CNESSI: Citation Network with Enhanced Structural and Semantic Information

Figure 3.1: Overview of methodology, showing integration of various phases through the development of CNESSI.

Since these sections from previous phase are 'raw', meaning they are not normalized and are specific to the document. As they can vary vastly across all documents, they need to be normalized into finite Generic Parts. Each research document generally contain sections talking about 'Motivation' (S1), 'Background' (S2), 'Literature Review' (S3), 'Methodology' (S4), 'Results and Discussion' (S5) and 'Conclusion and Future Work' (S6); these S1 to S6 are referred as 'Sections' (Generic Parts, Normalized sections). Hence, phase 3 contains supervised machine learning model which is responsible for normalizing the raw-sections by classifying each raw-section into one of the generic-parts. Henceforth, all the raw-sections of documents are now classified as one of the generic-parts. This phase is also responsible for closed citation resolution, where a 'Citation' is identified at each cite-code, with its location belonging to the respective section and its context. The metadata aids in resolving the cited-to document. In each document, all the content of all raw-sections that are classified as one class of generic-parts are merged, so that we have documents with generic-parts (normalized sections; referred from this phase onwards, simply as sections), where each generic part has consolidated information of all raw-sections.

The first three phases form a pipeline, while the phase 4 is independent phase, responsible for identifying the key-terms which represent a domain of knowledge (KT - Knowledge Topic). Since the dataset is limited to ACL research, following 5 different Knowledge-Topics are defined: 'NLP - Natural Language Processing' (KT1), 'IR - Information Retrieval' - (KT2), 'ML - Machine Learning' (KT3), 'DAA - Data structures and Algorithms' (KT4), 'HPC - Systems and High Performance Computing' (KT5). For each KT, respective conference names, in turn all the research papers of these conferences are acquired. The combined sets of keywords and the words in the paper titles form the terms for each KT. Normalizing the term frequency distribution for each KT, the KT-term probability distribution is derived.

In phase 5, experimentation and analysis is carried out to determine the unique traits or characteristics of each section. For each section, the respective text of every document are taken together as single text and is standard preprocessed (as shown in Figure 3.8), hence is represented by combined set of key-phrases counts and bag-of-words [27] - together referred as bag of phrases. Note that for each term; Document Frequency (df) (across different texts in this particular section across different documents) is considered rather than Term Frequency (tf). The Section-term df distribution is row-normalized to get Section-term probability distribution.

In phase 6, the term frequency (tf) for every term for every text chunk (text of every section and every citation context) is derived. Each text chunk is is standard preprocessed (as shown in Figure 3.8), hence is represented by combined set of key-phrases counts and bag of words - together referred as bag of phrases. This Text chunk-term frequency distribution is factorized using LDA to get Text chunk-topic and topic-terms probability distributions.

The KT-term probability distribution is used along with outputs of phase 5 and phase 6 to derive respective KSV (i.e., corpus level generic part trait representing KSV and KSV for each text chunk (consolidated information of each section of each research document along with every citation context of every research document, respectively)). Note that the phase5 and phase 6 are independent but follows the phase 4 and the initial pipeline of phases 1, 2 and 3.

3.2 *Phase 1: PDF to Python Objects with Raw Un-validated Sections*

This phase takes set of research documents in raw PDF format and converts it into python objects [18]. The flow phase1 has been shown in Figure 3.3. Python Objects are chosen as the base for data representation as it can be organized in Object Oriented Design. Moreover, it helps in easier manipulation and management of different structural components compared to other possible formats like xml. Different classes and their organization is depicted in Figure 3.2. Each document contains unique id, title, venue where it was published, year of publish, abstract, number of sections, number of cite-codes, list of authors, list of raw-sections and list of sections as its attributes. Each raw-section or a section is an object of class 'Section' which contains identifier - the identifier that occurs beside the section name in a research document, position - the index of the section in the research document, position normalized - position divided by number of sections in the research document, list of figures, number of paragraphs, list of paragraphs - where each paragraph is a long string, number of cite-codes, number of fig-references, number of sentences per paragraph, number words per paragraph, the generic part as which it is classified during the phase 3, generic part, generic part probabilities -

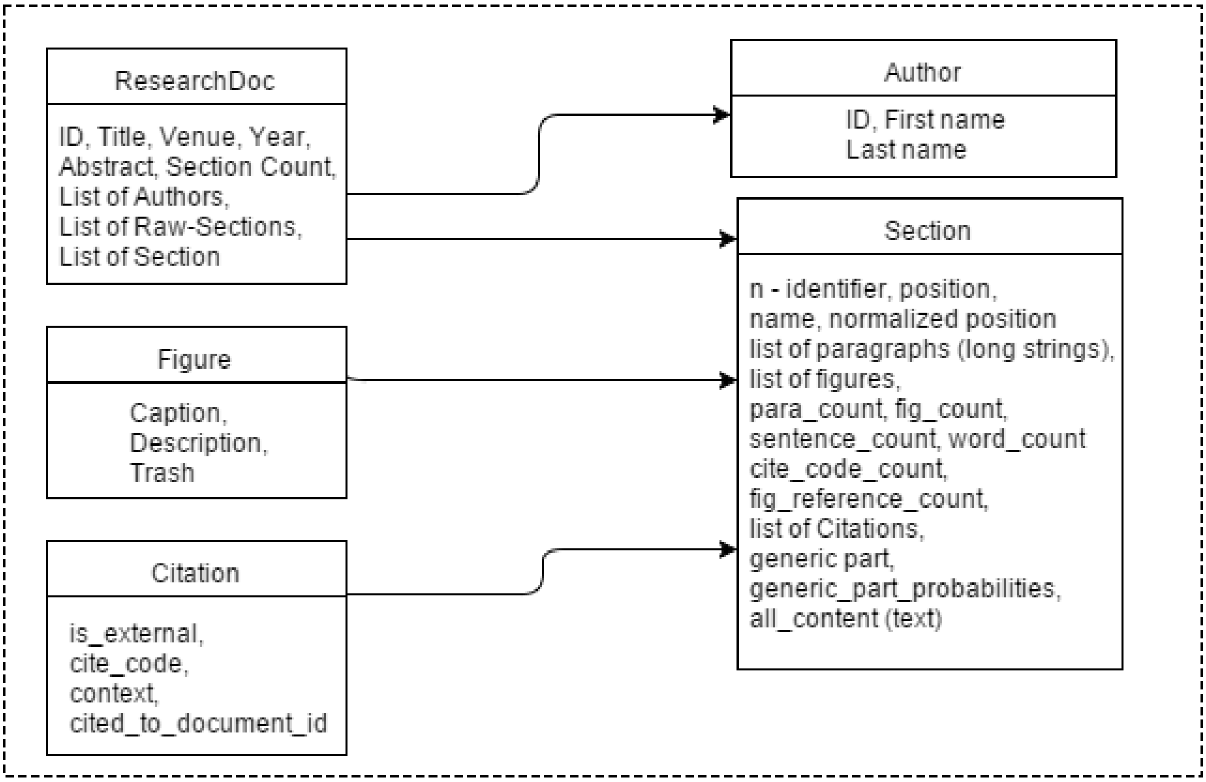

ResearchDoc
ID, Title, Venue, Year, Abstract, Section Count, List of Authors, List of Raw-Sections, List of Section
Author
ID, First name
Last name
Section
n - identifier, position, name, normalized position
list of paragraphs (long strings), list of figures, para_count, fig_count, sentence_count, word_count cite_code_count, fig_reference_count, list of Citations, generic part, generic_part_probabilities, all_content (text)
Figure
Caption, Description, Trash
Citation
is_external, cite_code, context, cited_to_document_id

Figure 3.2: Class diagram showing various attributes of Python Object representation of Research Documents.

probabilities with respect each generic part given by classifier in phase 3, all-content - all the text content in paragraphs and figures merged together. Each Figure object contains the fig caption, description given for some figures, and figure trash - some text identified after image processing on the figure. Each Citation object contains a is-external - flag indicating whether it is external citation or not, it also contains typical cite code, and context of this citation in this section which will be filled in phase 3. Clearly, the above explained python object representation of the data is more structured as each section and it attributes add structural information.

There are many inconsistencies in reading a PDF file, since it is not meant for computer processing. A PDF document could have been constructed from Microsoft Word, Latex or even with raw images. Hence extracted text from PDF by itself is a major problem. As addressed by many previous works regarding text extraction from PDF, GROBID [14] has been used in this phase, as it is State-of-the-Art in the section heading extraction (as discussed in Literature Review and Background chapters). Focus on section heading is important in this research as sections of a research paper are the main

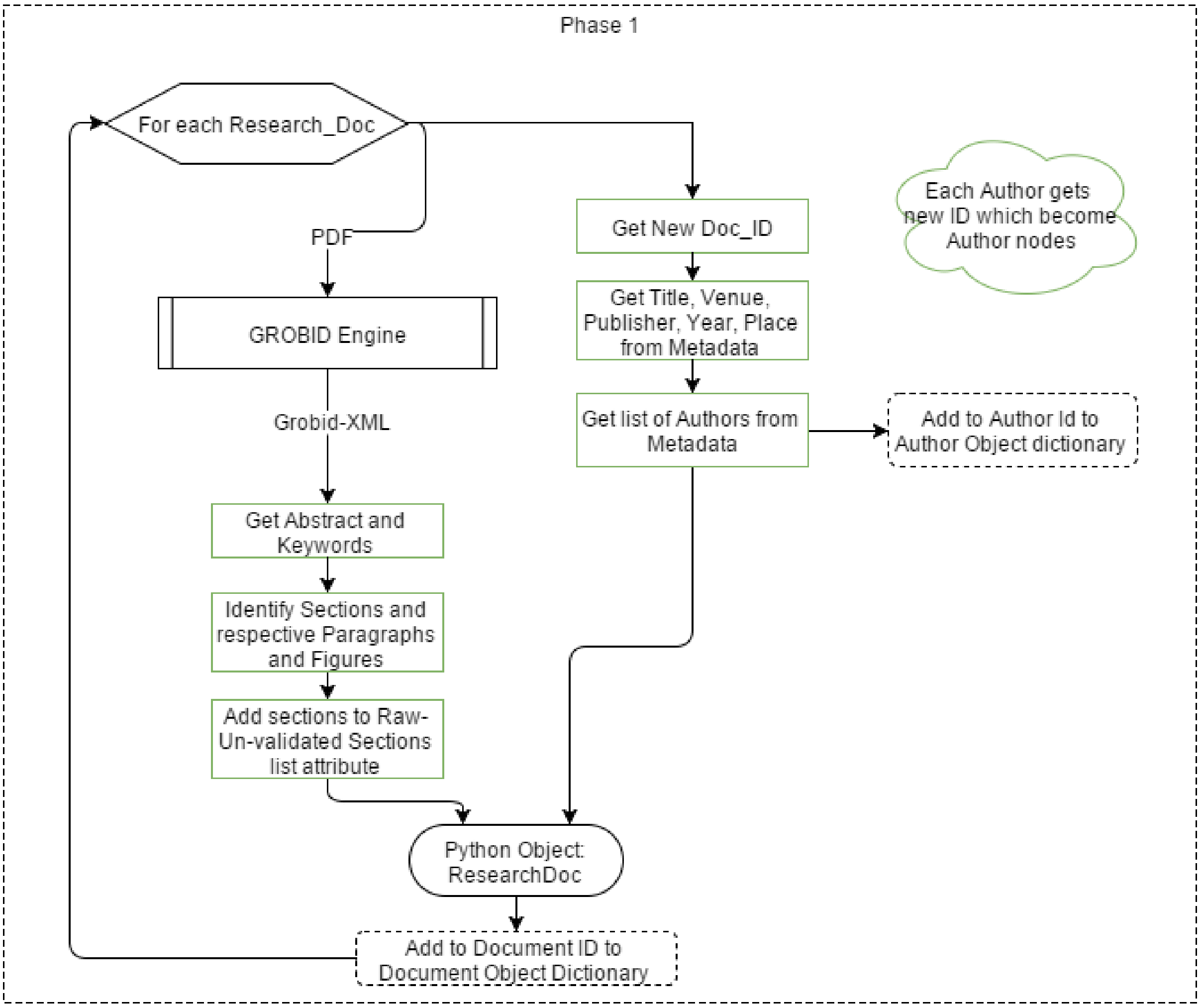

Phase 1
For each Research_Doc
PDF
GROBID Engine
Grobid-XML
Get Abstract and Keywords
Identify Sections and respective Paragraphs and Figures
Add sections to Raw-Un-validated Sections list attribute
Get New Doc_ID
Get Title, Venue, Publisher, Year, Place from Metadata
Get list of Authors from Metadata
Each Author gets new ID which become Author nodes
Add to Author Id to Author Object dictionary
Python Object: ResearchDoc
Add to Document ID to Document Object Dictionary

Figure 3.3: Description and data flow in Phase 1.

structural components to which different parts are the text are related. Each research document in PDF format is processed using GROBID engine which produces an xml file. Certain non-ASCII (ASCII [26]) characters are removed and in-line tags are normalized for proper parsing. GROBID created xml provides most of the structural information necessary, like the raw-section name and different paragraphs of raw-sections and figures etc.. But these raw-sections extracted need validation because during raw-section heading identification, some bold or italicised text could be mistaken for section heading. Hence these section extracted by GROBID are referred to as raw-un-validated sections (raw, because they are not normalized across corpus, which is done in phase 3 where they are classified into one of the generic parts).

The XML version of all research documents are then parsed and converted into python object with respective attributes. In addition, new document ID is given to each document and each research document object is hash mapped with respect to its ID. Similarly each unique author is hash mapped with respect to unique ID given to every author. These hashmaps basically form the nodes of the citation network. Through the research process, more and more semantic information are being added into the objects stored as values in the hashmap, thereby enhancing the semantic and structural information.

The dataset or the corpus contains 10951 ACL research documents and metadata with respect to each research document. But note that the same methodology can be applied to any dataset (which contains metadata; or after extracting metadata). These are in PDF format which cannot be structurally read by machines. Hence, it needs to be converted to more consistent readable format, like Python Objects [18]. (Implementation details are put up along with shared code [19].)

3.3 *Phase 2: Rule Based Raw Section Validation*

Moreover, observing the section names that were extracted through GROBID, we found lot of mis-identifications of section names. Hence this lead to the introduction of section validation phase. The reasons for invalid section name extractions have been discussed in the discussion section's Data organization and section validation subsection.

Therefore in order to solve these section name issues, a Rule-Based Algorithm for section validation is designed as shown in Figure 3.4. Here each Raw Un-validated section names are checked against certain rules which determine if it is valid section or

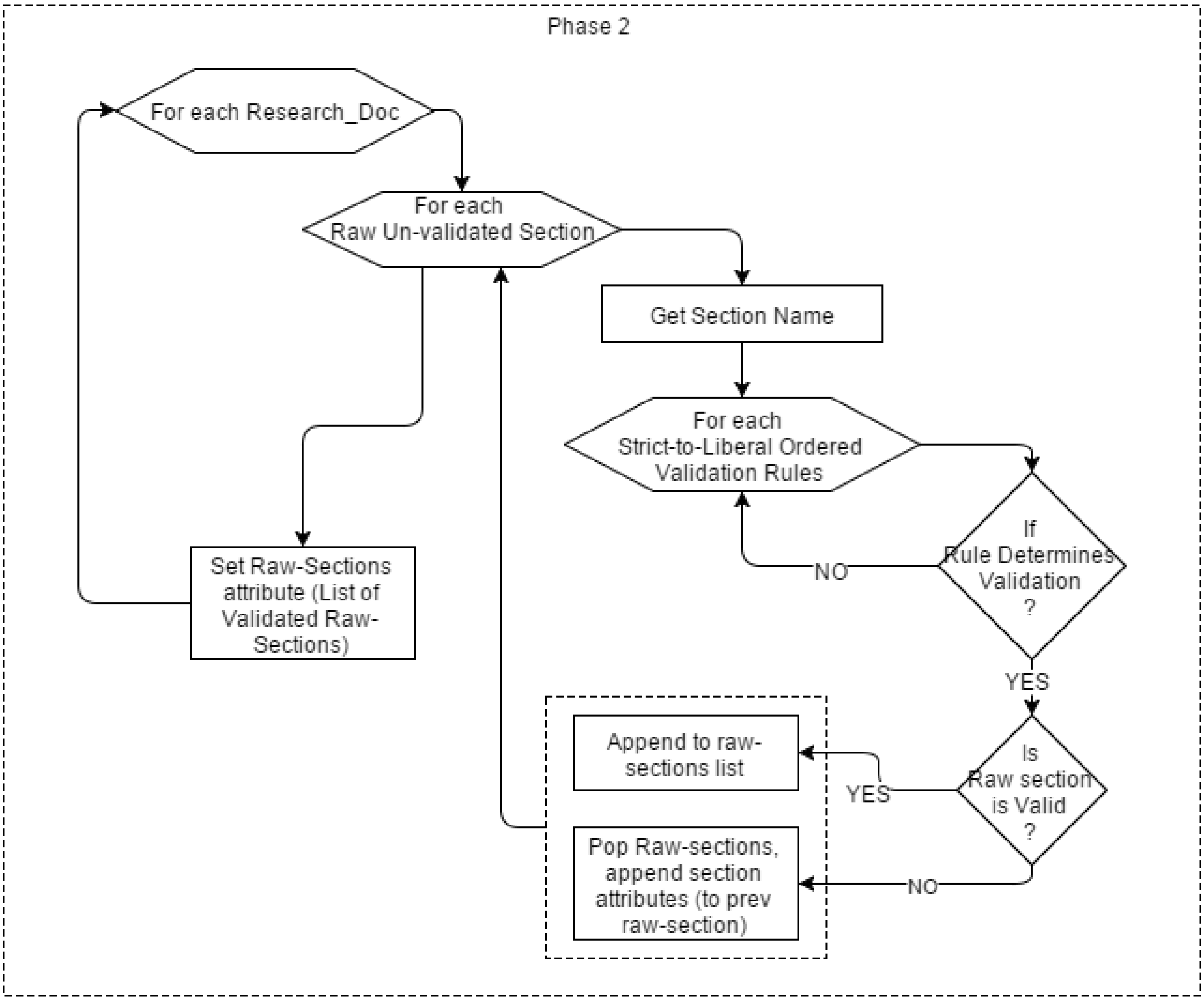

Phase 2
For each Research_Doc
For each Raw Un-validated Section
Get Section Name
For each Strict-to-Liberal Ordered Validation Rules
If Rule Determines Validation ?
NO
YES
Is Raw section is Valid ?
YES
NO
Append to raw-sections list
Pop Raw-sections, append section attributes (to prev raw-section)
Set Raw-Sections attribute (List of Validated Raw-Sections)

Figure 3.4: Description and data flow in Phase 2.

not. Some of the ordered rules shown in Table 1 were designed to tackle each of the issues. These rules are ordered in descending order of strictness.The rules were tuned by carrying out multiple trials. In each trial the frequency of matches against each rule were analysed for its correctness, and tuned to maintain a balance between precision and recall.

Table 1: Ordered Rules for Raw-Sections Validation

ID	Validity	Rule Description
1	FALSE	junk-null
2	FALSE	para-too-long
3	FALSE	fig-first
4	FALSE	junk-onlydigit
5	FALSE	junk-onlystop
6	TRUE	only-standard-exact-standard-name
7	TRUE	only-standard-edit-standard-name+edit_dist
8	TRUE	only-stemmedstandard-exact-standard_name
9	TRUE	only-stemmedstandard-edit-standard_name+edit_dist
10	FALSE	junk-nonenglish
11	FALSE	junk-onlynonupper
12	TRUE	title-all
13	FALSE	fig-somewhere
14	TRUE	identifier-exists
15	TRUE	title:
16	FALSE	junk-misc

Standard names of sections are identified by considering top fifty frequent section names having all words in English vocabulary. In this paragraph each rule is briefly explained starting from the first. 1: The section name contains only white space, then it is invalid. 2: Section name is longer than ten word, then it must be an extract from a paragraph, so it is appended to paragraph of previous section. 3: The first word in the section contains the word 'figure' or 'table', hence an extra figure count is added to the previous section. 4: Section names are categorized as junk - which is comprised only one word which is a digit are invalid. 5: Section names are categorized as junk - which is comprised only one word which is a common stop word are invalid. 6: Section name containing only word which exactly matches with a standard name is valid. 7: Same as above but edit distance of twenty percent with any standard name allowed. 8: Same as third rule but word stems were matched. 9: Same as fourth rule but word stems were matched. 10: Section names containing add words which are not present in English vocabulary are invalid. 11: Section names where none of the letters are capital are invalid. (title in the rule refers to initial letter being capital) 12: Section name is valid if all the words are initial letter capital. 13: If section name has 'figure' or 'table' as a sub string then it is invalid. 14: If the section identifier exists along with the section name then it is valid. 15: If the section name has a word having initial letter capital followed by a colon then section name is valid. 16: Section not caught up in any of the above rules are considered invalid.

All the content in the invalid raw-sections of a document are merged into the previous valid section. The reason for doing so is described in the Results and Discussion sections Data Organization and section validation subsection. If the first raw-section itself turns out to be invalid, then its content its put into new raw-section named 'Introduction', based on a general assumption that generally every research document has Introduction as its first section. After this phase, the documents have raw-section which have been validated. (Implementation details are put up along with shared code [19].)

3.4 *Phase 3: Classification of Raw Sections into Generic Sections*

Since the raw-sections from previous phase are 'raw', meaning they are not normalized and are specific to the document. As they can vary vastly across all documents, they need to be normalized into finite Generic Parts. Each research document generally contain sections talking about 'Motivation' (S1) and Introduction, 'Background' (S2), 'Literature Review' (S3), 'Methodology' (S4), 'Results and Discussion' (S5) and 'Conclusion and Future Work' (S6); these S1 to S6 are referred as 'Sections' (Generic Parts, Normalized sections). Hence, phase 3 contains supervised machine learning model which is responsible for normalizing the raw-sections by classifying each raw-section into one of the generic-parts. The flow in phase 3 has been pictorially described in Figure 3.5.

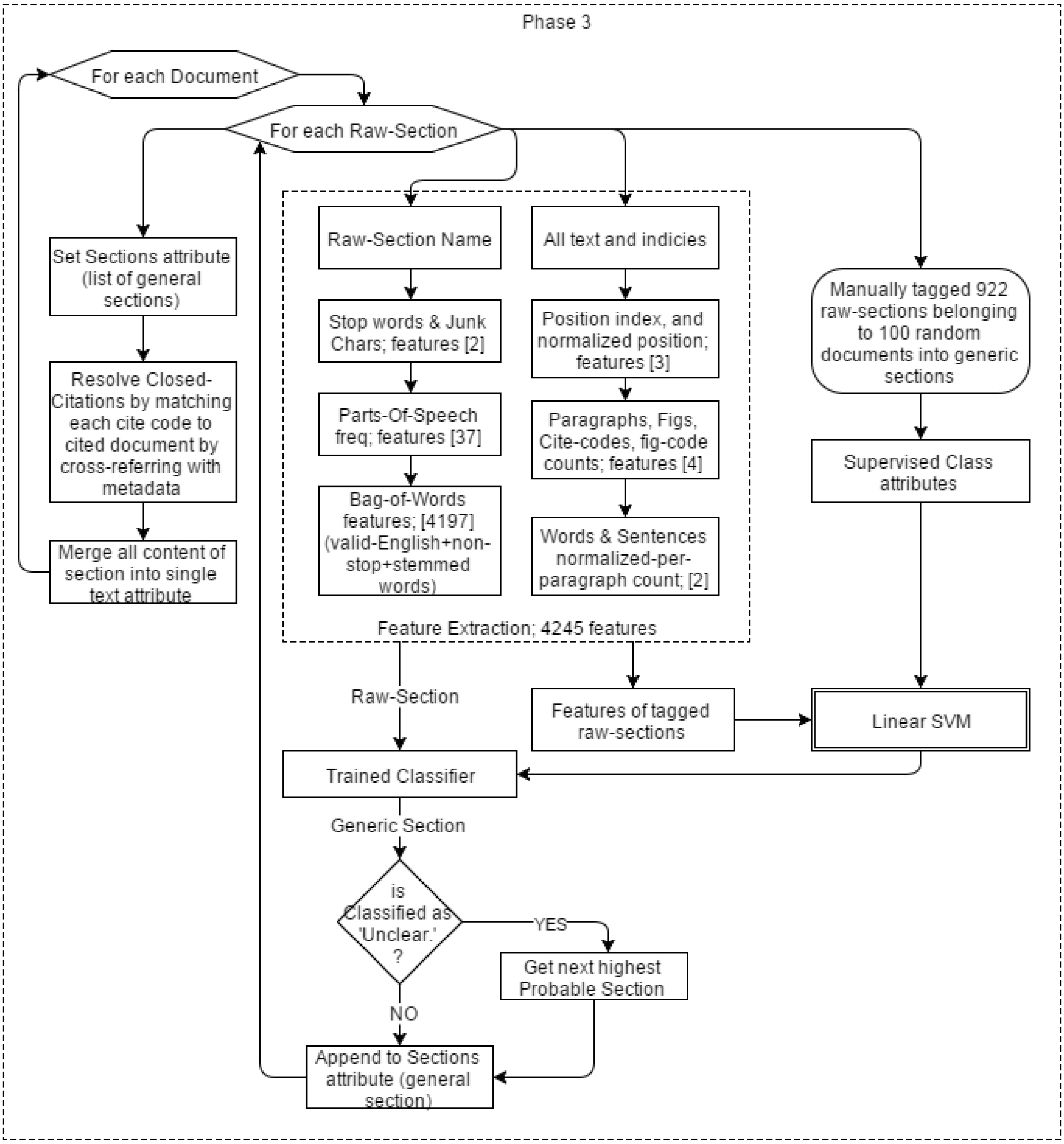

Phase 3
For each Document
For each Raw-Section
Set Sections attribute (list of general sections)
Resolve Closed-Citations by matching each cite code to cited document by cross-referring with metadata
Merge all content of section into single text attribute
Raw-Section Name
Stop words & Junk Chars; features [2]
Parts-Of-Speech freq; features [37]
Bag-of-Words features; [4197] (valid-English+non-stop+stemmed words)
All text and indicies
Position index, and normalized position; features [3]
Paragraphs, Figs, Cite-codes, fig-code counts; features [4]
Words & Sentences normalized-per-paragraph count; [2]
Feature Extraction; 4245 features
Manually tagged 922 raw-sections belonging to 100 random documents into generic sections
Supervised Class attributes
Raw-Section
Features of tagged raw-sections
Linear SVM
Trained Classifier
Generic Section
is Classified as 'Unclear.' ?
YES
Get next highest Probable Section
NO
Append to Sections attribute (general section)

Figure 3.5: Description and data flow in Phase 3.

Since supervised classification needs training data, 100 documents were chosen at random and their respective 922 raw-sections were manually tagged by looking at their section name and by looking into the PDF document as one of these Generic sections. Some section were unclear to tag, hence were tagged as '?' (Unclear). These form the supervised training set for classification.

- *Feature Extraction:* Each raw-section is represented through a set of Positional and structural features from each raw-section along with textual features of its name. Nine Positional and structures features are chosen; Raw-section's position, position normalized with the number of position, section identifier, number of cite-codes, number of figure-references, number of figures, number of paragraphs, number of sentences per paragraph and number of words per paragraph. These positional and structural features might be characteristic to each generic section and have been chosen hypothesising that they might help in classification.

  Textual features from the section names are extracted as follows. Firstly, miscellaneous features like stop-words count and junk characters count are found. Stop-words [40] are the words like is, the, are, etc. which occur with high frequency in any text, which in turn might add noise to several analysis but sometimes help in characterizing authoring behaviour.Then, the raw-section names are preprocessed using standard NLP preprocessing techniques like removal of stop-words, Porter stemming [21] and removal of words not present in English vocabulary. Stemming makes sure that various morphological forms of a word are considered to be the same.

  Once the preprocessing is done, parts-of-speech based features are extracted. NLTK [2] python module contains standard NLP tools. NLP's default parts of speech tagger has a tag-set with 37 different parts-of-speech [36] considering context based pats of speech variation as well. The counts with respect to each of 37 parts-of-speech for each raw-section name are extracted. In addition, bag-of-words [27] is evaluated for raw-section names which totalled to 4197 different words. Bag of words implies calculating the word frequency without taking the order of occurrence of words into account.

  The feature vector ultimately comprises of 4245 features, with 9 positional and structural features, 2 miscellaneous features, 37 parts-of-speech based features and 4197 bag-of-word features. Each raw-section of every document is represented with such feature vector.

The supervised set (containing 922 samples of raw-sections manually tagged as one of the 6 generic sections or as '?' marking 'unclear' class) represented through above described diverse set features.for classification is now represented through quite diverse set of features. By training a supervised machine learning model on this training set, will help in identifying the patterns in the raw-section which make them belong to one of the generic-sections. It is a supervised multi-class classification problem. Since the feature is set is quite large and diverse, SVM [41] is hypothesised to work well. The experimentation comprises of Linear SVM with different values of penalty parameter 'C' for the error term [24]. The experimentation was run on Linear SVM as a 5-fold cross validation on the training set with values of C varying from 0.1 to 2.0. Cross-validation [30], sometimes called rotation estimation, is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

After the cross-validation results, the values of C which gives the best results is chosen for final classification model. Once Linear SVM is trained with the best value of C obtained in the cross validation on the complete training set, we get a trained model. This is used to classify the rest of the untagged samples into generic sections. The trained model gives probability of classification against each generic-section for each sample of raw-section. The generic section having the highest probability is considered as the generic section to which the raw-section belongs. In case a raw-section is classified as '?' indicating its 'Unclear', then it is put to the generic section having next highest probability.

In each document each raw-section is now put into one of the 6 generic sections.

Hence, all the content in the raw-section belong to one particular generic section are merged in each document separately (this will set the 'sections' attribute of each research doc object). If no raw-section is found to belong to some generic section in a document, then that generic section in this document will have no information, then this document is said to not contain that generic section. As an output of this phase, each document is structured to comprise all or some of the 6 generic sections (each having their information). Note that during merging raw-section, the information like the text in paragraph is not lost, hence it can be called as loss-less merging. From this phase onwards, the generic sections are simply referred to as sections for ease of explanation.

Before the loss-less merging of raw-sections, the cite-codes are resolved. Here, each cite code is looked-up into the metadata containing closed citations and identified it this cite-code is referring to a particular document, by matching the pattern for author names and published year that is present in the cite-codes. Cite-codes are looked up for the pattern using the reglar expression like -

'([(]?[ ]*AuthorLastName[ ]*(et al.)?[,]*Year[ ]*[)]?)'

where AuthorLastName and Year checked for all documents present in the closed citation found in metadata. If it matches, then it can be said that this particular cite code is referring to that document (called as cited-to-document) from this section, hence using this information a Citation object is appended to the 'citations' attribute of this raw-section. The citation context of this citation is considered to be 5 sentences above and below the sentence in which this cite-code is found, but this range doesn't span paragraphs. Meaning if the cite-code occurs in last but one sentence of a paragraph then 5 sentences before the cite-code existing sentence and only 1 sentence after the cite-code is considered as the paragraph ends there. This is done by the general understanding that the subject change across paragraphs, hence the context of citation context should be considered only within the paragraph which contains the cite-code. Similarly each citation in every research document is resolved and then loss-less merging of raw-section into sections is carried out. At the end of this phase, each document is structurally represented as sections (generic sections; normalized sections).

The classification has been implemented using Scikit-Learn [17] module or python. SVM.SVC module is used which can give the probabilities with respect to each class

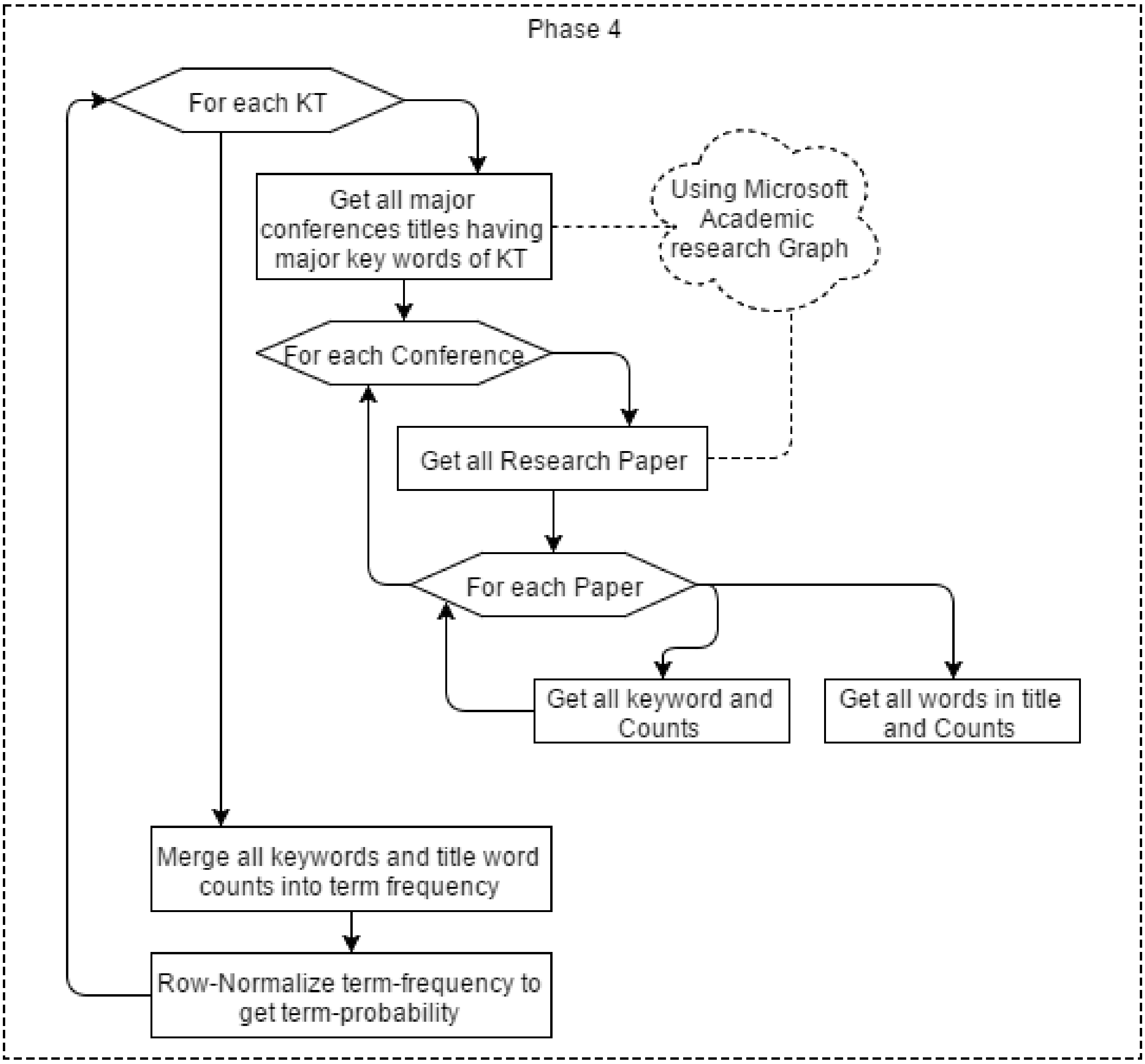

Phase 4
For each KT
Get all major conferences titles having major key words of KT
Using Microsoft Academic research Graph
For each Conference
Get all Research Paper
For each Paper
Get all keyword and Counts
Get all words in title and Counts
Merge all keywords and title word counts into term frequency
Row-Normalize term-frequency to get term-probability

Figure 3.6: Description and data flow in Phase 4.

after classification but normalizing the distance from hyperplanes. It provides options for cross-validation, randomizing the samples and setting different values of C. Once the model is trained, it can classify the un-tagged data in order to complete the supervised classification of each raw-section into generic sections.

3.5 *Phase 4: Knowledge Topics-Terms Mining*

The first three phases form a pipeline, while the phase 4 is independent phase, responsible for identifying the key-terms which represent a domain of knowledge (KT - Knowledge Topic). Since the dataset is limited to ACL research, following 5 different Knowledge-Topics are defined: 'NLP - Natural Language Processing' (KT1), 'IR - Information Retrieval' - (KT2), 'ML - Machine Learning' (KT3), 'DAA - Data structures and Algorithms' (KT4), 'HPC - Systems and High Performance Computing' (KT5).

If a human who is considered expert in a KT, say NLP, is asked to read two research papers and say which has more knowledge in KT, say NLP, his ind would look for the number of significant NLP terms in both documents and he would weigh different terms with the amount of significance that particular term has with respect to NLP. His mind would have implicitly derived these weights as the NLP expert would have read most of the research documents pertaining to NLP and would have observed the frequency and impact of these terms.

The aim in this phase is perform similar intellectual decision to derive knowledge-term probability matrix, where each KT is represented in a probabilistic distribution of various terms based on their frequency in research documents of major conferences pertaining to the respective KT. This has been achieved as shown in the Figure 3.6

Microsoft Academic research Graph (MAG), has a database with huge amount of research papers with their keywords and the metadata information of the conference they have been published in. The Microsoft Academic Graph [20] is a heterogeneous graph containing scientific publication records, citation relationships between those publications, as well as authors, institutions, journals and conference "venues" and fields of study.

For each KT, we need to identify significant terms which represent this knowledge domain. Initially, certain major ubiquitously known phrases that would appear in the conference names of each KT are collected manually. All the conferences which have the any of these well-known phrases in the conference names are identified for each KT. All the papers of each conference for each KT are found by looking into MAG. Further, all the keywords and respective frequency counts of each paper for a particular KT are found. Former keywords counts catch all the major keywords the authors have identified on the papers on the KT, which might have most of the common phrases and words belonging to KT. But some new terms could have been coined in the titles of the papers belonging to a KT, which needs to be caught. Hence, the words in the titles (title-words) and their frequency counts of each paper for a particular KT are found. The keyword-frequency-counts and title-words-frequency-counts are merged and together referred as term-frequency (tf) for each KT. In order to treat each plural and singular forms of every

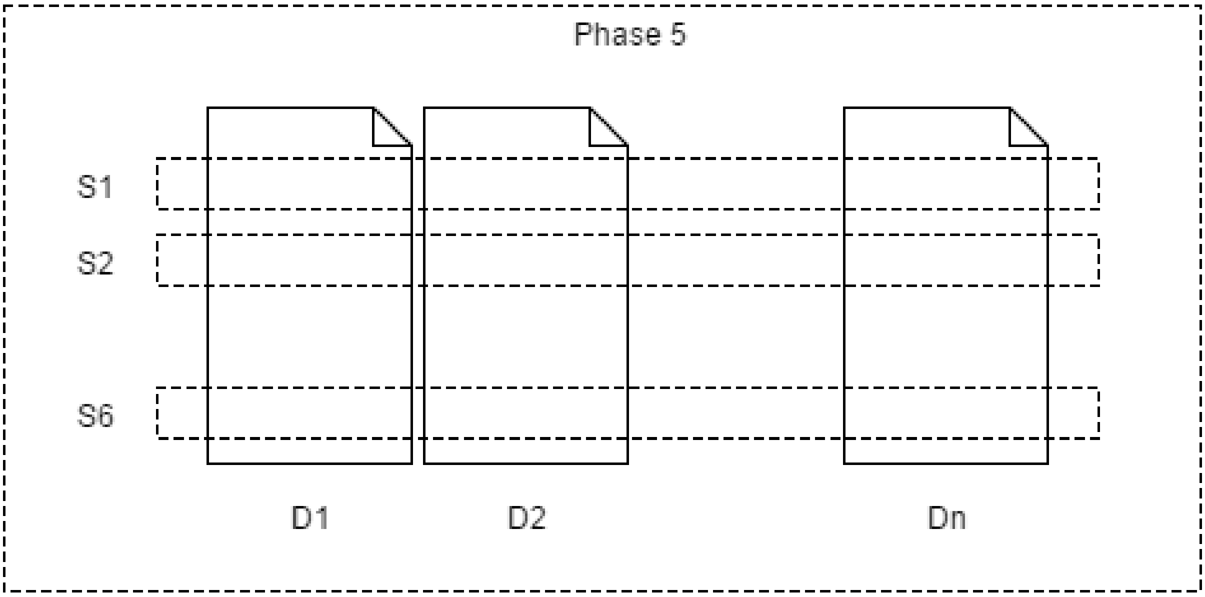


Figure 3.7: Description and data flow in Phase 5.

word equally, these terms were singularized using pattern [6] and the counts were merged respectively. Further, these are row normalized to get KT-Words (KT-W" matrix; where W" represents the terms list) matrix as shown below.

$$\begin{array}{c c} \underline{\textbf{KT-W"}} & \begin{array}{cccc} w"1 & w"2 & \dots & w"p \end{array} \\ \begin{array}{c} KT-1 \\ KT-2 \\ \vdots \\ KT-5 \end{array} & \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots \end{pmatrix} \end{array}$$

For each KT, respective conference names, in turn all the research papers of these conferences are acquired. The combined sets of keywords and the words in the paper titles form the terms for each KT. Normalizing the term frequency distribution for each KT, the KT-term probability distribution is derived. (Implementation details are put up along with shared code [19].)

3.6 *Phase 5: Section-wise Trait Analysis*

In phase 5, experimentation and analysis is carried out to determine the unique traits or characteristics of each section. Note that this analysis is done at the corpus level and not per research document. The aim aim is to find the general representation of each section

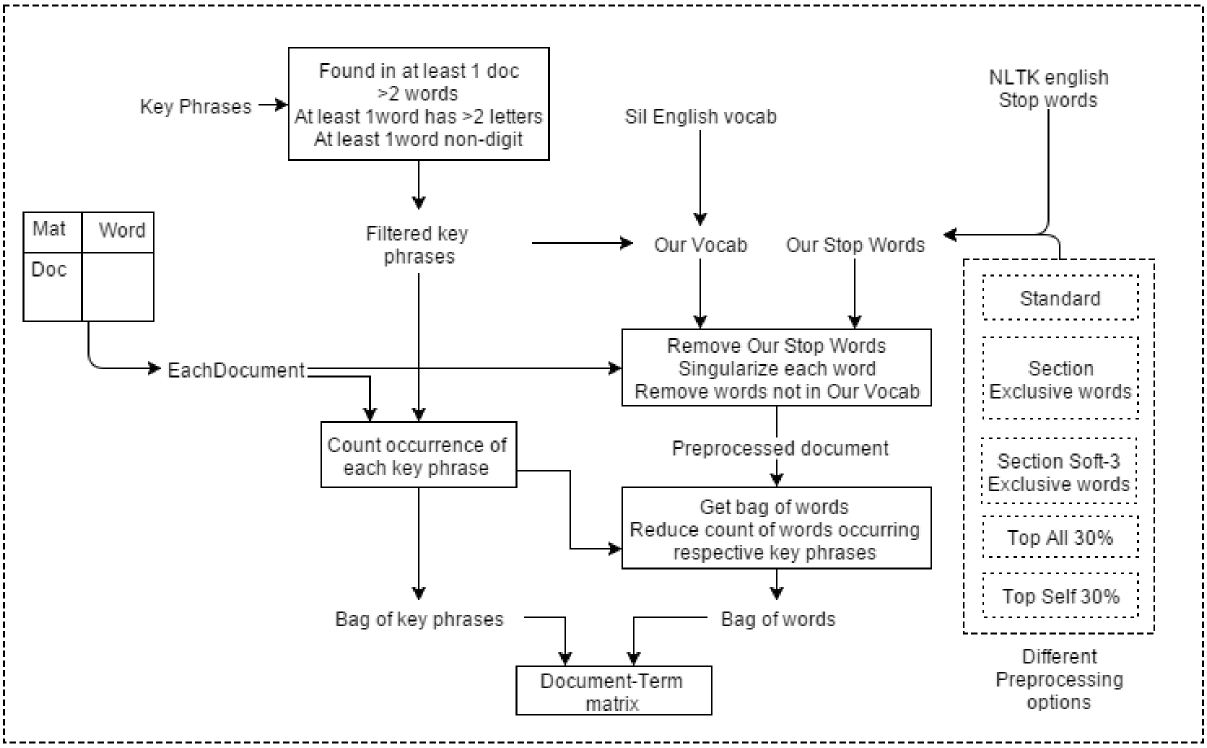


Figure 3.8: Text preprocessing flow for document word matrix.

across the corpus. Hence, find the the general characteristics of knowledge share of each section across whole corpus taken as one.

Note that LDA is used to factorize a document-word matrix into document-topic and topic-word matrices. In each case, document with reference to LDA inputs are differ. The text being represented as word against the documents in document-word matrix are standard processed as shown in the Figure 3.8 (right hand side). Standard processing shown refers to processing where only standard English stop words are used with no custom stop words. Other ways of processing are done in Phase 5 for experimentation purposes as explained in the Results and Discussion chapter, only to study the variation of topics characteristic to each generic part with varying set of stop words.

For each section, the respective text of every document are taken together as single text (as shown in 3.7) and is standard preprocessed (as shown in Figure 3.8), hence is represented by combined set of key-phrases counts and bag-of-words [27] - together referred as terms. Note that for each term; Document Frequency (df) (across different texts in this particular section across different documents as shown in 3.7) is considered rather than Term Frequency (tf). The Section-term df distribution is row-normalized to get Section-term probability distribution which can be represented by the Section-Words matrix (S-W' matrix; where W' represents the terms list) shown below.

$$\begin{array}{l} \underline{\textbf{S-W'}} \\ S-1 \\ S-2 \\ \vdots \\ S-6 \end{array} \begin{array}{c} \begin{array}{cccc} w'1 & w'2 & \dots & w'q \end{array} \\ \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots \end{pmatrix} \end{array}$$

- *General Traits KSV at Corpus level:* KSV at the corpus level which represents each Section in the Knowledge Topic space (KT space), implying the general distribution of knowledge share of each section across whole corpus. This can be derived by using the KT-W" matrix and S-W' matrix. In order to normalize the words list that represent the KTs (W") and Sections (W'), the columns with respect to all the words of W" that are not present in W' are deleted from KT-W". Also, zero columns are added for all the words that are present in W' but not present W" in KT-W". Now KT-W" gets normalized to KT-W'. Using below equation wherein these matrices and multiplied by taking transpose for compatibility, we can get Section-KT matrix.

$$(S \times W') * (KT \times W')^{\mathrm{T}} = S \times KT \tag{1}$$

The Section-KT matrix represents each Section in the Knowledge Topic (KT) space, implying the general distribution of knowledge share of each section across whole corpus. The Section-KT matrix would like the below matrix.

$$\begin{array}{l} \underline{\textbf{S-KT}} \\ S-1 \\ S-2 \\ \vdots \\ S-6 \end{array} \begin{array}{c} \begin{array}{cccc} KT-1 & KT-2 & \dots & KT-5 \end{array} \\ \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots \end{pmatrix} \end{array}$$

This is the matrix provided at the Corpus level as specified in the Problem Statement chapter as shown in Figure 2.3. It can be observed that the Citation Network now contains corpus level Semantic information on how Knowledge is shared generally across sections. (Implementation details are put up along with shared code [19].)

3.7 *Phase 6: LDA on all Sections & Cite Contexts of all Documents*

In Phase 6, the aim is to represent each text-chunk (all the sections and all the citation contexts of all research papers are referred together as text-chunks) as a term frequency distribution, hence derive KSV for each text chunk. This is basically representing each text-chunk in the KT space.

In phase 6, the term frequency (tf) for every term for every text chunk (text of every section and every citation context as shown in Figure 3.9; The D11 to Dn6 are text chunks of sections of all documents as shown in the image, whereas D1,c1 to Dn,cm are the text chunks of all citation contexts of all documents. For ease of representation, all of these together are represened as D1 to Dz; by refering the total number of text chunks to be z) is derived. Each text chunk is standard preprocessed (as shown in Figure 3.8), hence is represented by combined set of key-phrases counts and bag of words - together referred as terms.

The Text-chunk-term frequency matrix looks as shown in below matrix.

$$
\begin{array}{cc}
\underline{\textbf{D-W}} & \begin{array}{cccc} W1 & W2 & \dots & Wr \end{array} \\
\begin{array}{c} D-1 \\ D-2 \\ \vdots \\ D-z \end{array} &
\begin{pmatrix}
\dots & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots \\
\vdots & \vdots & \vdots & \vdots \\
\dots & \dots & \dots & \dots
\end{pmatrix}
\end{array}
$$

This is factorized using LDA to get Text chunk-topic and topic-terms (Topics are denoted by ’T’) probability distributions as shown in following two matrices. (LDA was run to extract (soft cluster) to factorize into 100 topics processed through 500 iterations).

$$
\begin{array}{cc}
\underline{\textbf{D-T}} & \begin{array}{cccc} T-1 & T-2 & \dots & T-100 \end{array} \\
\begin{array}{c} D-1 \\ D-2 \\ \vdots \\ D-z \end{array} &
\begin{pmatrix}
\dots & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots \\
\vdots & \vdots & \vdots & \vdots \\
\dots & \dots & \dots & \dots
\end{pmatrix}
\end{array}
$$

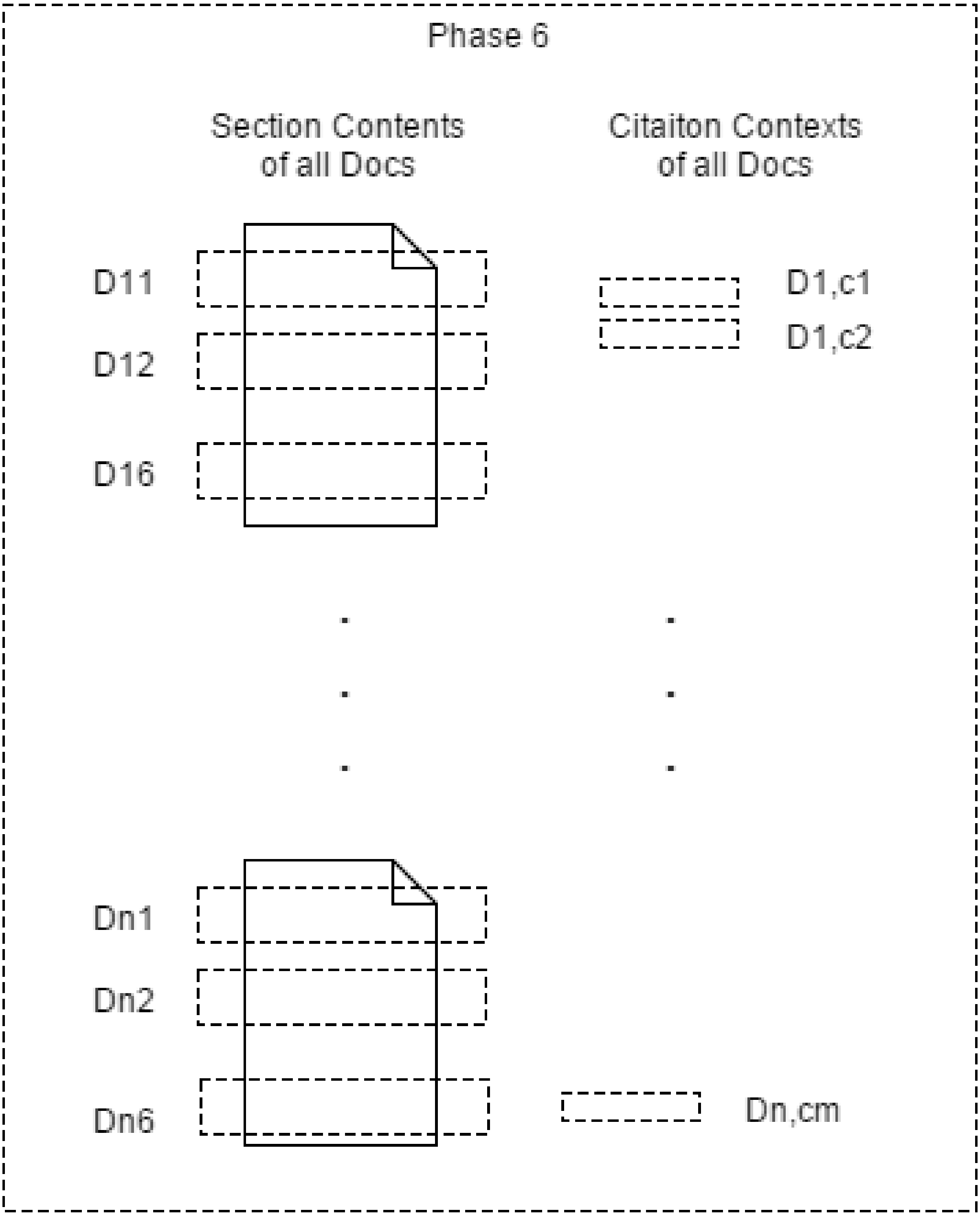


Figure 3.9: Description and data flow in Phase 6.

$$\begin{array}{l} \underline{\textbf{T-W}} \\ T-1 \\ T-2 \\ \vdots \\ T-100 \end{array} \begin{array}{c} \begin{array}{cccc} W1 & W2 & \dots & Wr \end{array} \\ \begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots \end{pmatrix} \end{array}$$

- *KSV for each Text-chunk (nodes and edges):* KSV for each Text-chunk which represents each Text-chunk in the KT space, implying the specific Knowledge share of each text-chunk. This can be derived by using the KT-W" matrix, D-W matrix and T-W matrix.

  Firstly, consider KT-W" matrix and T-W matrix. In order to normalize the words that represent KTs and Topics, the columns with respect to all the words of W" that are not present in W are deleted from KT-W". Also, zero columns are added for all the words that are present in W but not present W" in KT-W". Now KT-W" gets normalized to KT-W. Using below equation wherein these matrices and multiplied by taking transpose for compatibility, we can get Topic-KT matrix.

  $$(T \times W) * (KT \times W)^{\mathrm{T}} = T \times KT \tag{2}$$

  Now, consider the Text-chunk-Topic (D-T) matrix and Topic-KT (T-KT) matrix. These two can be multiplied as shown in below equation to get Tech-chunk-KT (D-KT) matrix.

  $$(D \times T) * (T \times KT) = D \times KT \tag{3}$$

The Text-chunk-KT (D-KT) matrix represents each Text-chunk (each section and citation context of every document) in the Knowledge Topic (KT) space, which represents each text-chunk specifically as a KT distribution. The D-KT matrix would like the below matrix.

$$\begin{array}{c|cccc} \underline{\textbf{D-KT}} & KT-1 & KT-2 & \dots & KT-5 \\ \hline D-1 & \dots & \dots & \dots & \dots \\ D-2 & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D-z & \dots & \dots & \dots & \dots \end{array}$$

Considering the text chunks belonging to sections, their respective KTs form the semantic information on each node (each research document) in the citation network, as specified in at the node in Figure 2.3 present in Problem Statement Chapter. On the other hand, considering the text-chunks of citations, the respective KTs form the semantic information on each edge of the citation network, as specified in at the node in Figure 2.3 present in Problem Statement Chapter. Looking at the Citation Network created now, it can be observed that each research document has enhanced structural information as it has general-section wise information, as well as enhanced semantic information as it has KSVs on each section and each citation context. Hence, all the objectives specified in the Problem Statement chapter are completed. Hence, the Citation Network with Enhanced Structural and Semantic Information (CNESSI) is created. (Implementation details are put up along with shared code [19]. Since the matrix sizes can get really huge and processing time is quite large, it was run on cluster with multiple-cores and memory as high as 128GB.)

4 Results and Discussion

This sections contains most of the discussion around the results obtained and analysis of various observations. Initially the dataset used in the experimentation and implementation is described.Following which the first two phases have been discussed, followed by each subsection discussing each of the rest of the phases.

4.1 *Dataset Description*

A dataset with reliable metadata would be a better choice. Hence, ACL-Arc dataset [1] is a corpus of scholarly publications about Computational Linguistics. This corpus is a canonicalized subset of the ACL Anthology, up to February 2007, consisting of 10,921 articles, which is meant be used for benchmarking applications for scholarly and bibliometric data processing. The PDF files for each of the research document is available along with metadata regarding their ID, title, Authors, Venue. ACL Citations metadata is used to build basic citation network where each research document node has the rawPDF file along with it.

4.2 *Data Organization and Section Validation*

After PDFs were converted to GROBID-XML, there were parsing errors and cite-code could not be recovered due to inline tags used. Hence substituting the inline ¡ref¿ tags with (#ref#) tags helped to overcome this minor issue.

Some PDFs were built through images, hence GROBID failed to extract textual information from it. Such documents were kept-out of our dataset. Hence out of 10591 documents, 7995 were considered for further study while the rest were discarded.

Moreover, observing the section names that were extracted through GROBID, we found lot of miss-identifications of section names. Hence this lead to the introduction of section validation phase. The reasons for such miss-identifications were analysed by looking into the documents where such errors were occurring. Sections that spanned the columns, sections with headers and footnotes on the first page were the some of the section identification issues inside GROBID. Other issues were, bold and italicised non-section headings, and figure captions being identified as section names, all capitalized words in between the paragraphs etc.. The nature of these issues reveals that there is a higher recall, meaning more sub-strings are identified are section names than the true section names while there are less section names that have been missed out. This observation added the necessity to add section validation before section classification.

The rules mentioned in the phase2 subsection of methodology section were designed to tackle each of the issues discussed above. These rules are ordered in descending order of strictness. The rules were tuned by carrying out multiple trials. In each trial the frequency of matches against each rule were analysed for its correctness. Similarly, the order of the rules were tuned to improve clarity of section validation. Initially rules lead into high recall version, then adding strict rules lead into high precision. Henceforth, the order was adjusted to have a balanced version of section validation.

Observing the nature of invalid sections, it was decided to merge the attributes of invalid sections to the previous valid section. For examples, after a valid section was identified, say, some bold text in the section itself was identified as another section. In fact, the latter section contents belong the former section. In order to make such correction, once the section validate identifies the latter section as invalid, then its sections could be merged with the former valid section. After processing 7995 research documents as explained, 68804 raw-sections were identified as valid sections.

4.3 *Classification of Raw Validated Sections into Generic Sections*

Cross validation results of SVM classifier to classify every raw-section into generic-section are analysed to identify the best value of C suitable for the dataset as discussed in Phase3 of methodology chapter. The below table 2 shows the accuracy score for different values of C.

The Mean accuracy must be higher and mean variance must be lower for in order to achieve better classification. Observing the above table 2, we can clearly see that the accuracy slowly increases when C is changed from 0.1 to 0.5 and starts decreasing when C is increased from 0.5 to 2.

The Mean accuracy of 73% doesn't look convincing. But it is to be noted that one of the classes fed into SVM is '?' to refer to unclear raw-sections even during manual tagging. Hence, we need to analyse the confusion matrix to understand the trend of

Table 2: Cross-Validation results of SVM Classifier with different values of C to classify each raw-section into one of the generic sections

C	**Mean Accuracy (%)**	**Mean Variance (+/-)**
0.1	70	0.05
0.2	71	0.06
0.3	71	0.06
0.4	73	0.05
0.5	*73*	*0.05*
0.6	73	0.06
0.7	72	0.06
0.8	72	0.06
0.9	72	0.07
1.0	71	0.05
1.5	71	0.07
2.0	69	0.07

miss-classifications.

Analyzing the confusion matrix of SVM with value of C being 0.5 shown in Table 3 it can be analysed that the trend in the miss-classifications. We see that most of the miss-classifications are for the class '?' which was unclear even during human tagging, hence these miss-classifications can be neglected. Looking at other miss-classification, it can be seen that there only positional miss-classifications, for instance, the Methodology sections have been classified as results and vice-verse since they are found consecutively in most of the research papers, but the Motivation sections has never been misclassified as results since they never occur together. In fact, in most of the research documents, sometimes part of the results are mentioned in the methodology sections, hence these miss-classifications are not harmful. Similarly, Lit Review and Background sections are mixed up, which in fact is sometimes true. By looking at the positional inertia here, it can be understood that the positional and structural features have contributed to such represented which has favoured the research. Hence, this classification is considered legitimate and from this phase onwards all the information in each research document is

Table 3: Confusion matrix for SVM with C=0.5 to analyse the trend in miss-classifications

Classified as =>	**0 (?)**	**1**	**2**	**3**	**4**	**5**	**6**
0 (?)	*14*	0	1	0	5	0	0
1	0	*19*	0	1	3	0	0
2	0	1	*0*	0	5	0	0
3	0	0	1	*4*	4	0	0
4	10	0	0	0	*82*	13	0
5	2	0	0	0	13	*27*	1
6	3	0	0	0	3	1	*18*

structural organised as the sections (referred as generic sections from this phase onwards).

4.4 *Knowledge Topics-Terms Mining*

Table 4: KT and representative conferences, papers and terms

KT-ID	**KT**	**Conferences**	**Papers**	**Title-words**	**Keywords**	**Terms**
1	NLP	33	30367	14081	7231	21312
2	IR	16	6999	5691	3031	8722
3	ML	77	63740	21768	13176	34944
4	DAA	28	12110	7534	4200	11734
5	HPC	37	16087	1058	6896	7954

The Table 4, shows the number of terms with respect to each KT as well as the number of conferences and papers they were extracted from. These were obtained as described in the Phase 4 of Methodology chapter. All the KT together are represented by probability distribution across a set of 48129 terms.

4.5 *Section-wise Trait Analysis*

Experimentation to analyse the traits of each section, initially involved considering text in all documents pertaining to each section separately. Followed by standard processing as shown in Figure 3.8 by considering the term frequency (tf), which is then input to LDA to extract 20 topics processed through 500 iteration.

The topics-words were analysed and it could be clearly observed that lot of frequent words were overshadowing the words which actually represent each section characteristically. Hence, various different kinds of input preprocessing were tried out as shown in Figure 3.8; Section exclusive words - considering only those words which are exclusive for the section. Section soft 3 exclusive words - Considering words that occur in not more than 3 other sections. Top all 30% - Top frequent 30% words in other sections are removed from the current section under consideration. Top self 30% - Top frequent 30% words in the current section under consideration are excluded.

All these prepossessing showed minor improvements, but could not characteristically represent each of these sections. Hence, these experiment lead to the strategy shift to consider texts across different documents of one particular section in all documents form one row. And such rows of all sections were together processed under LDA with 100 topics processed through 500 iterations. This clearly showed significant improvements in bringing up the top topic words which represented each section characteristically as shown in the Appendix I. This lead to the decision of representing section data in such manner could be semantically useful.

But, LDA factorization must lower the rank, else it would not add additional benefit and document frequencies would be sufficient and efficient while representing each section. Hence only the df was considered to represent each section. All the sections were represented with 38402 terms. Hence, the KSV for each section at Corpus level (the S-KT matrix) as shown in the Phase 5 subsection of Methodology chapter was derived.

This is the matrix provided at the Corpus level as specified in the Problem Statement chapter as shown in Figure 2.3. It can be observed that the Citation Network now contains corpus level Semantic information on how Knowledge is shared generally across sections.

4.6 *LDA on all Sections & Cite Contexts of all Documents*

All the text-chunks were represented through 39735 terms, hence were fed into LDA. LDA being unsupervised model can only be subjectively evaluated or evaluated through proxy. The rest of the methodology contains clear matrix multiplications where correctness is implicitly understood.

Some of the top 20 words of first 10 topics appearing for text-chunks are shown below:

- *Topic 0:* language, natural language, natural, system, ha, linguistic, research, application, understanding, problem, knowledge, natural language processing, processing, paper, nlp, many, is a, work, human, technique

- *Topic 1:* list, item, input, new, token, match, step, first, output, block, set, two, one, filter, is a, contain, pas, original, generated, sequence

- *Topic 2:* speech, recognition, system, speech recognition, speaker, recognizer, using, rate, wa, spoken, error, used, error rate, word, performance, acoustic, utterance, vocabulary, transcription, datum

- *Topic 3:* classification, label, classifier, instance, class, task, set, prediction, classified, category, example, two, classify, individual, et, combination, naive, one, different, feature

- *Topic 4:* summary, document, article, news, summarization, story, information, topic, sentence, extract, source, text, content, abstract, extraction, original, title, newspaper, using, email

- *Topic 5:* parser, parsing, parse, sentence, tree, grammar, treebank, constituent, parsed, use, input, using, penn, output, syntactic, partial, lr, used, statistical, result

- *Topic 6:* datum, program, database, file, code, computer, data base, base, query, information, part, format, wa, form, stored, field, command, may, version, time

- *Topic 7:* word, unknown, context, dictionary, vocabulary, used, is a, character, frequency, two, sequence, first, following, occur, window, part, one, size, list, since

- *Topic 8:* error, correct, correction, word, repair, case, wa, incorrect, spelling, example, detection, failure, wrong, false, due, result, input, problem, type, correctly

- *Topic 9:* template, slot, fill, type, text, th, object, key, incident, message, score, wa, subtotal, analyst, location, response, entity, set, mi, id

- *Topic 10:* probability, estimate, distribution, datum, given, is a, model, parameter, estimation, estimated, pr, likelihood, using, probabilistic, use, used, smoothing, training, frequency, count

Further, as explained in the phase 6 of Methodology chapter, KSV for each text-chunk (the D-KT matrix) was found. Considering the text chunks belonging to sections, their respective KTs form the semantic information on each node (each research document) in the citation network, as specified in at the node in Figure 2.3 present in Problem Statement Chapter. On the other hand, considering the text-chunks of citations, the respective KTs form the semantic information on each edge of the citation network, as specified in at the node in Figure 2.3 present in Problem Statement Chapter. Looking at the Citation Network created now, it can be observed that each research document has enhanced structural information as it has general-section wise information, as well as enhanced semantic information as it has KSVs on each section and each citation context. Hence, all the objectives specified in the Problem Statement chapter are completed. Hence, the Citation Network with Enhanced Structural and Semantic Information (CNESSI) is created.

4.7 *Implementation Details*

Specific implementation details with reference to the code written during the research is provided in here with respect to phases 1 and 2. the python script document-classes.py contains the class definitions for each of the python objects described above as shown in Figure 3.2. The python script grobid-batch-run.py's grobid-run-batch() method processed each PDF into GROBID-XML. The xml-preprocess() method replaces the inline ”ref” tags referring to cite-codes and fig-codes to non xml based custom tags and removes non printable characters, to facilitate clear xml parsing through ElementTree module in Python. The metadata-preprocess() method replaces misplaced HTML statements in the metadata file that came along with the dataset in ACL-ARC. The script parsing-pdf.py organize-inputs() methods organizes each xml with respect to each pdf file after conversion from GROBID along with the respective metadata file from ACL ARC metadata. Further the process-inputs() method parses each xml using ElementTree module, by extracting various attributes of documents and raw-unvalidated sections as shown in the Figure 3.2. The above method delegates the task of validating the raw-section to test-section-name()

(which in turn has several organised in the order or their strictness into basic and advances set of rules each catered by separate auxiliary methods)which implements the rule based section validation algorithm discussed in phase 2. The process-inputs() method also takes care of loss-less merging of invalid section sections into previous valid section. If the first section is invalid, then it names it 'Introduction' and marks it as first valid section. Further, section-text-preprocess() method extracts few positional and structural features pertaining to each section which will later be used during classification in phase 3. This returns a preprocessed XMLs for all documents.

Phase 3 involves tagging for sections into one of the generic parts in order to create supervised training set. The script section-tag-organizer.py has write-section-tag() method which writes a csv files to help the human tagger and write-positional-features() method which writes the positional features of each section into a numpy pickled file (features.npy). The section-classification.py script has prep-ml-input-all() method which used the features.npy to which parts-of-speech and bag-of-words features are appended with the help of bow-preprocessor() method, hence it saves the feature vectors with respect to each section in X-all.npy (It has been designed to save each intermediary file and then input it to next step, so that memory can be cleared after each step allowing such big files to be executed on a commodity machine). The run-input-finalisation() method takes the X-all.npy and creates the supervised training data that can be fed to SVM (it uses the get-tagged-section() method for acquiring the tagged the samples for training data). The classifier-experiment() method does 5-fold cross validation with SVM with parameter C values ranging from 0.1 to 2.0. It uses the Scikit-Learn [17] module which contains method for implementation cross validation for SVM and outputs respective results. The value of C for which best results are obtained are then applied to the SVM and trained on the whole training set (in classifier-selected.py). Then the model which is fit on the training set is use to classify the rest of the data (method for this task are available on Scikit-Learn module. Using an additional parameter the scikit-learn SVM model can output the classification probabilities with respect to each class as well.). The post-classification-key-fix() method fixes the document ID and section index as they are the primary keys for each row representing a unique section. The citation-fixation.py script has a method citation-fixation() which uses extract-cit-from-text() which resolves each cite-code (for each cite-code, with respect to every cited document for current document as available in the metadata, it checks if this cite-code contains the authors last name and year, if so this cite-code it said to be resolved as its cited document id is found), and identifies the citation context. The traversal is made by first dividing the text (text in paragraphs, figure captions and description) into sentences using nltk [2] python package, and 5 sentences above and below and the sentences where cite code exists without spanning paragraphs is considered to the citation context. Advance check for citation is also done, to catch the cite-code which were not identified by the GROBID in the initial phase. To enable this, the cite codes already established are matched using regular expression and they are deleted from the text, followed by matching for pattern of each authors last name and year for every possible cited document. The test-all-meta-citation-found() method checks if every closed citation as indicated in the metadata are found in the text or not (indeed, some closed citation cite-code were not found). All the instances of the same citation occuring in the same section are merged into one Citation object. The method post-classification-object-fix() sets the 'generic-part' attribute of every Section object of respective Research-Doc object as classified in this phase. Further, the post-classification-gen-part-consolidation() method merges sections classified as the same generic-part in the same document. If classified as unclear '?', then it takes the next higher probability having generic-part and merges with respective section. Hence the Citation Network with structural information is ready by the end of phase 3.

Phase 4 is independent of previous 3 phases, hence can be implemented before or after implementing previous phases. This phase uses Microsoft Academic research Graph (MAG). The knowledge-terms-mining.py script contains each of the following methods for retrieving the keywords with respect to each of the knowledge-topics. Five knowledge topics mentioned in the abbreviations as they are the main knowledge domains covered in the dataset ([1]) chosen. The prep-kt-keywords-dict-1() method prepares a dictionary of conferences with respect to each KT by matching sub-strings of major words with respect to each KT from the list of conference titles available in MAG (It creates the inverse dictionary as well). The prep-kt-keywords-dict-2() method prepares a dictionary with KT mapped to respective list of conference papers which belong to respective list of conferences (It creates the inverse dictionary as well). The prep-kt-keywords-dict-3() prepares a dictionary of keywords and their counts with respect to each KT by taking the keywords mapped to each conference papers of each KT. The prep-normalized-kt-keywords-dict() merges the counts of keywords after singularizing the words in each keyword (singularized by pattern module). The dictionary is realized as a matrix with each row being a KT and the keywords being the columns (the unique keywords of all KT taken and their counts appended). Similarly, the prep-kt-title-words-dict() prepares matrix of KT versus counts of words in titles of conference papers which were mapped to each KT (they are singularized). The prep-kt-merged-count-dict() method creates a matrix by merging both the matrices from KT-keywords and KT-title words to make it KT-terms matrix. The The matrix is row-normalized to represent the KT as a probability distribution of keywords (by prep-kt-prob-mat() method).

Phase 5 include processing of text and making the term counts matrix with respect to each generic part as shown in Figure 3.8. The final-ksv-pre.py script contains methods to create standard list of keywords and vocabulary.The prepare-my-vocab() method takes the standard English vocabulary from sil [7]. Then it takes all the keywords (phrases) in MAG, and adds the words in keywords that are not there in standard English dictionary and together is considered as my-vocab (This is our vocabulary which includes scientific words other than standard English words). The prepare-our-phrases-all() method takes all the keywords in MAG (Phrases) then considers only those which are existing in our dataset (by looking in all the text in our dataset). This time expensive task is required to reduce the size of matrix in further steps.In prepare-our-phrases-filtered() method these are validated using simple rules like each word in the phrase must not be digits etc. and only valid phrases are considered as set of our-phrases. Lets call removing all words other than my-vocab and the default list of stop words in nltk [2] as standard preprocessing. In phase 5, all the text of each section from all documents is taken separately for each section, then document frequency taken against all the phrases found in previous step. Then standard preprocessing is done, after which document frequency is taken for all the words in my-vocab. This matrix has 6 rows and the number of columns is equal to the count of phrases and my-vocab together. Since it is huge sparse matrix is maintained. (Care is taken to ensure a word is not counted twice. i.e. if a word is counted while matching against a phrase, that word is not counted again while taking counts against that word. In order to reduce such counts, sparse matrix would be very expensive, hence it is converted to dense matrix for this calculation, then it is converted back to sparse matrix. For the conversion from sparse to dense matrix Scipy module in python is used.

For sparse matrix, csr (compressed sparse row) sparse matrix is used which is available in Scipy [12].) This matrix is combined with the matrix in phase 4 as explained in the phase 5 in Methodology. Such matrix multiplication of huge matrices is carried out using Numpy [22] module in python. In order to save these matrices and other objects in python the Pickle module for object serialization is used. This gives the matrix at the corpus level.

Phase 6 involves standard preprocessing and matrix creation as similar to phase 5 implementation explained above, but all the text of all section and the citation contexts of all documents are considered separately. Hence the number of rows here is equal to total number of sections and citation contexts in the whole corpus. Such a huge matrix is maintained as sparse. But for count reduction it is converted to dense as explained above in phase 5 implementation. This definitely does not fit into the memory of commodity machines. Hence it was run on a cluster with 128GB memory. Hence this matrix after converting to sparse is given as input to LDA (lda package in Python) which outputs a fit-model containing the factorized matrices as shown in Phase 6 of methodology section. Hence is combined with the matrix obtained in phase 4 as explained in Phase 6 in methodology. This is the matrix at each node and edge of the citation network.

5 Conclusion and Future Direction

The research started out with raw citation network with raw PDF documents. The hybrid methodology involved different phases each involving keen natural language processing using machine learning and algebraic and statistical techniques, which ultimately created a Citation Network with added Structural and Semantic Information (CNESSI) (as specified in the problem statement). CNESSI widens the horizon of Citation Network analysis to uncover profound insights hidden among the scholarly articles.The Future Research and Applications discussed below clearly emphasises the high scope and usability of CNESSI as a enhanced citation network.

5.1 *Future Work*

CNESSI opens up plethora of research opportunities in semantically superior citation network analysis. The analysis with CNESSI are not restricted to mere counts and frequency based analysis. CNESSI provides structural and semantic knowledge represented through the KSV which are present as additional information at each node and edge of the Citation Network (as shown in the matrices in Figure 2.3). In this Chapter some of the semantic analysis that could be done with CNESSI are discussed. This would help the readers to understand the usage possibilities of CNESSI.

1. *KSV of a Research Document:*

   CNESSI provided structurally broken down KSV with respect to each Section of document as the matrix provided at each node representing a research document. Further KSV of the complete research document can be derived, which can be semantically interpreted as the knowledge share of the document.

   Trivially, summing the KSVs of all sections would give the KSV for the research document. But a particular text being written in Motivation section and being written Methodology section are interpreted differently when humans read it. Hence in order to imbibe such section biased interpretations, the KSV of sections could be summed up with weights for each section $(w_1, w_2, ..., w_6)$ that add the Section specific bias. The weight can be calculated using the below equation with respect each section of a particular document.

$$W_i = K_{si} \cdot K_{gi} \tag{4}$$

Where i is the generic section ID and W_i is the weight for section i. K_{si} is the KSV for section i in this document level present at the node representing the current research document. K_{gi} is the KSV of section i at the corpus level, where generic traits of the section in the corpus is represented. The sum weighted summation (using the above derived weights) of KSV of section would give us the KSV of the document.

2. *Cited-to Section Identification:*

   Citation Networks now contain information regarding documents which have cited each other. But the the information like which section of the cited document was actually cited from citing document, which is essentially tells us which section in the cited document inspired the citing document authors to cite it.

   For a citation in the citing paper, the similarity scores of the citation context can be taken for each of the section in the cited document. The section in the cited document which gives highest similarity score can be considered as the cited section (or inspiring section) in the cited document (Since the vectors are finite dimensional, simple cosine similarity [29] would be sufficient). The value of the similarity score can quantify the amount of inspiration taken during this citation as well. Perhaps, by matching the KTs, how much inspiration across each KT can be quantified. This clearly is a very high quality information which can describe the citation network as a inspirational network and illustrate the citation network with high semantic information. The structural information derived from this kind of analysis would help to understand how each section of a research document inspired different sections across all the citing documents.

3. *Knowledge Projection based Ranking:* Nowadays, the ranking of research documents is mainly done taking the citation counts. But all citations are not equal [43]. Clearly the citation where the citing document has taken more information from the document must be relatively given a higher weight. Hence, the documents could be ranked based on these weights which represent amount of knowledge that is taken from the cited document.

In order to do such ranking, instead of counting the number of citations, where each citation is considered as one (equal), the scalar product [31] of KSV of citing and cited document can be considered which represents the projection of knowledge of the cited document on the citing document. (The KSV for each document can be retrieved as shown in the 'KSV of a Research Document' list item in this enumerated list). Ranking based on the score which is sum of these projection of knowledge for each citation for a particular paper can be debated to be a superior since it takes the semantic representation of the research documents into account.

The above list shows the different application and analysis that could be done with CNESSI. Also the methodology to use the additional structural and Semantic information available in CNESSI is illustrated. It should be noted that, while using the values in KSV, one should be ensure to normalize it by comparing against other KSVs. A single KSV independently cannot be interpreted, as it is made for relative and comparative semantic analysis of different research documents.

Some of the other analysis which could be completed with the use of CNESSI are listed below.

- *Author-to-Author Domain analysis:* Matching the KSV of documents any pair of authors have worked on reveals the information regarding the type of collaboration each pair of authors have. Similarly the knowledge domain concentration of any research group can be determined. In addition, the impact of each research group with respect to different domains of knowledge can be determined.

- *Knowledge Diffusion and Knowledge Evolution:* Determine the knowledge domain which is generally more prevalent at citations, can in turn help in deriving the knowledge domain is being diffused relatively at a higher rate. Henceforth, mapping the year when these documents were published we can find how each knowledge domain has been evolving through time.

- *Author-specific Focus Analysis and Author Impact Ranking:* An author might have published various res each documents throughout his research career. By comparing the semantic information with each of his research documents in the order of their published year, the shift occurring in his focus knowledge domain through his career can be tracked. Such author specific analysis done for every author, helps to rank authors per knowledge domain. This could help anew research find appropriate authors who are focused in a particular knowledge domain. This in turn can help to improve the search for high impact authors in each knowledge domain which can serve as a superior metric for ranking authors compared to H-Index and its variations [5].

CNESSI could be further improved by expanding the scope of knowledge domains, and the research documents taken into consideration. More efficient text extraction from PDF format could improve the quality of the research subjectively. Through effective and comparative evaluation of applications like the ones described above, can reveal the usefulness of CNESSI. Such proxy evaluations can reveal options to improve the methodology to derive CNESSI.

This work work mark significant changes in the way current research documents and authors are being ranked and to analyse the knowledge flow in the research documents. The interesting results that could be obtained through semantic analysis on CNESSI could reveal profusion of insights of paramount significance.

6 References

[1] Steven Bird. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. 2008.

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] Lutz Bornmann and Hans-Dieter Daniel. The state of h index research. *EMBO reports*, 10(1):2–6, 2009.

[6] DM Conway. An algorithmic approach to english pluralization. In *Proceedings of the Second Annual Perl Conference*, 1998.

[7] SIL International Linguistics Department. English wordlists, 2005.

[8] Leo Egghe and Ronald Rousseau. Introduction to informetrics: Quantitative methods in library, documentation and information science. 1990.

[9] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[10] Mark Girolami and Ata Kabán. On an equivalence between plsi and lda. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 433–434. ACM, 2003.

[11] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[12] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-12-03].

[13] Miray Kas. Structures and statistics of citation networks. Technical report, DTIC Document, 2011.

[14] Patrice Lopez. Grobid - documentation, 2015.

[15] Patrice Lopez. Grobid - information extraction from scientific publications, 2015.

[16] John Mingers and Loet Leydesdorff. A review of theory and practice in scientometrics. *European Journal of Operational Research*, 2015.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] Tutorials Point. Python object oriented, 2015.

[19] Bhuvan M S. Github-nessi, 2015.

[20] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 243–246. International World Wide Web Conferences Steering Committee, 2015.

[21] tartarus. The porter stemming algorithm, 2006.

[22] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[23] Anthony FJ Van Raan. Sleeping beauties in science. *Scientometrics*, 59(3):467–472, 2004.

[24] the free encyclopedia Wikipedia. What is the influence of c in svms with linear kernel, 2012.

[25] the free encyclopedia Wikipedia. Arnetminer, 2015.

[26] the free encyclopedia Wikipedia. Ascii, 2015.

[27] the free encyclopedia Wikipedia. Bag-of-words model, 2015.

[28] the free encyclopedia Wikipedia. Citeseer, 2015.

[29] the free encyclopedia Wikipedia. Cosine similarity, 2015.

[30] the free encyclopedia Wikipedia. Cross-validation, 2015.

[31] the free encyclopedia Wikipedia. Dot product, 2015.

[32] the free encyclopedia Wikipedia. Google scholar, 2015.

[33] the free encyclopedia Wikipedia. Latent dirichlet allocation, 2015.

[34] the free encyclopedia Wikipedia. Latex, 2015.

[35] the free encyclopedia Wikipedia. Levenshtein distance, 2015.

[36] the free encyclopedia Wikipedia. Part-of-speech tagging, 2015.

[37] the free encyclopedia Wikipedia. Portable document format, 2015.

[38] the free encyclopedia Wikipedia. Scopus, 2015.

[39] the free encyclopedia Wikipedia. Stemming, 2015.

[40] the free encyclopedia Wikipedia. Stop words, 2015.

[41] the free encyclopedia Wikipedia. Support vector machine, 2015.

[42] the free encyclopedia Wikipedia. Xml, 2015.

[43] Xiaodan Zhu, Peter Turney, Daniel Lemire, and André Vellino. Measuring academic influence: Not all citations are equal. *Journal of the Association for Information Science and Technology*, 66(2):408–427, 2015.

Appendix I: Specific Traits of each Section

This appendix contains analysis details of LDA output to analyse the traits of each section by considering texts across different documents of one particular section in all documents form one row. And such rows of all sections were together processed under LDA with 100 topics processed through 500 iterations. This clearly showed significant improvements in bringing up the top topic words which represented each section characteristically as shown below.

- 1. Motivation:

  semantic, determine, sentence, text, particular, datum, related, one, assigned, alway, appropriate, paper, include, result, another, need, even, information, use, position, based, system, syntactic, provide, make, source, two, current, also, state, much, take, way, certain, tile, linguistic, several, type, correct, possible, is is, node, main, used, handle, wa, found, may, time, is a, however, np, part, new, described, ha, problem, structure, must, term, task, word, language, algorithm, ref, could, shown, list, see, rule, large, rather, form, common, context, many, including, model, mean, similar, example, first.

- 2. Background:

  determine, sentence, defined, text, datum, let, related, one, assigned, appropriate, paper, include, result, another, need, consider, even, information, use, position, described, form, alway, provide, make, point, two, current, also, state, much, take, way, certain, np, new, several, type, correct, possible, is is, node, main, used, wa, found, may, respectively, end, is a, however, tile, the second, part, instance, linguistic, based, ha, problem, structure, must, ha a, term, task, word, language, algorithm, ref, could, shown, list, see, rule, large, rather, common, context, time, including, model, mean, similar, example, first.

- 3. Literature Review:

  semantic, statistical, *proposed, sentence, text, within, datum, al, related, one, paper, include, result, another, need, et, even, information, use, based*, system, approach, provide, make, technique, two, research, current, also, much, take, way, certain, linguistic, several, ref, main, particular, used, handle, wa, machine translation, may, component, is a, however, framework, part, *recent, new, described, ha, problem, structure, must, found, term, task, word, language, algorithm, reported, type, could,* syntactic, list, source, large, rather, good, computer, common, context, many, including, model, similar, example, first.

- 4. Methodology:

  consider, sentence, text, datum, assigned, alway, including, instance, the second, include, possible, tile, using, term, ha a, list, common, found, mean, another, related, see, result, even, shown, section, current, state, new, ref, correct, wa, respectively, given, however, let, could, context, first, point, one, appropriate, determine, use, described, three, indicate, much, way, type, form, part, ha, must, case, main, word, value, is is, following, similar, example, figure, defined, certain, need, information, end, provide, make, note, also, take, np, several, node, used, may, is a, two, structure, task, algorithm, rule, rather, time, position, model.

- 5. Results and Discussion:

  figure, sentence, text, *datum, related, one, paper, include, result, another, using, need, table, total, compared, best, based, even, information,* use, three, described, indicate, provide, much, make, show, precision, two, note, also, experiment, take, way, certain, problem, new, several, type, *common, main, used, wa, may, given, test set, error, is a, however, following, part, performance, e test, ha, linguistic, structure, must,* case, comparison, term, task, word, language, algorithm, ref, could, current, list, value, large, rather, shown, due, context, recall, training, including, model, section, similar, example, first.

- 6. Conclusion and Future work:

  useful, *author, sentence, partially, text, supported, datum, related, one, paper, include, result, another, need, even, information, use, like, based, system,* would, grant, provide, make, two, research, current, also, much, take, way, certain, main, linguistic, hi, several, ref, method, development, project, used, wa, may, helpful, is a, however, contract, part, new, using, ha, problem, structure, must, term, task, word, *algorithm, language, described, type, could, work, list, rule, large, rather, future, common, context, science, including, model, similar, example, first.*

Italicised are the exclusive words for taking top 10 topics and top 10 words from each topic. These exclusive words were found among this short range of words only in 3, 5 and 6 sections. In other sections its not found because of the skewed distribution of text chunks into sections. Motivation, Methodology and Results sections higher compared to other sections.

Here section 1's top 10 words from top 10 topics could be occurred very frequently in other sections as well since there significantly higher number of documents which covers many more words which might have present in other sections. Moreover, it is introduction which could be quite versatile.

Section 2 is Background, and section 3 is Lit Review, the latter having more number of documents could have covered up the words occurring in Section 2, causing no exclusive words to show up.

Section 4 being Methodology has lesser docs than Results i.e. Section 5. As we have seen the confusion matrix during classification miss-classification existed between two classes, causing only the section with higher documents to show up some exclusive words.

Nevertheless, when the number of top topics and words were increased to top 20 topics and top 100 words from each topic, following exclusive words were found for respectively shown sections:

- 1. Motivation:

  designing, motivate, *call, organized, logical form, ordinary, montague, overcome, reader, seek, traditionally, convenient, information need, search engine, quickly,* employ, characterize, background, popular, encoded, generally, interested, huge, outline, introduce, year, predefined, internet, advance, programming, structured, computational linguistic, n gram, sag, answering, objective, quantity, phrase structure, thompson, concern, review, concentrate, granted, syntactic analysi, largely, overview, central, briefly, portable, perspective, brief, concluding, challenging, subtask, hereafter, automatic speech, motivation, application domain, throughout, expensive, demand, assigning, conception, costly, f logic, obstacle, question answering, widely, formal, essential, mathematical, sketch, chomsky, *translate, presenting, henceforth, commercial, apo, fee, superior, appealing, as is, familiar, conclude, diversity, rapid, artificial intelligence.*

- 2. Background:

  violate, unmarked, guarantee, verify, enter, maximal, rt, seed, lion, stop, twice, merging, iv, io, ic, permutation, passive, deleted, obj, denoting, *nt, f value, asking, slower, trial, acceptable, er model, remember, non, el, artifact, threshold, regarding, propagation, locally, external,* versa, read, inter, putting, initially, arbitrarily, pu, unify, execution, synset, equivalence clas, proportional, vice, giving, *six, lh, merge, matrix, request,* inherited, ll, ce, finished, e function, satisfying, choosing, home, triggered.

Italicised words clearly represent these sections. Hence, it can be observed that by considering more number of topics and more number of top words from each topic, we would find exclusive words which represent the section with high semantic knowledge.

Still we dont find any specific terms for section 4. Hence, increased the top topic to 30, we find the following exclusive words:

- 4. Methodology:

  modal verb, m3, m2, london, fix, production, rp, rs, rn, r2, kleene, causing, indexed, n value, ws, filtered, sample size, appearing, substituted, *multiply, understandable, nc, push, unexpected, rood, cardinality, the net, map, liquid, seeing, nominal, e3, xi, clas p, traverse*, saw, subsequent, affecting, comp, 6a, mod, o1, governed, belonging, abc, sui, finite number, response, accumulate, harm, undergo, alpha, topmost, *exclude, disallowed, uniform distribution, iil, sire, sim, outlier, microphone, displayed, adjoined, ct, restore, wet, assuming, branch.*

Clearly italicised words are those used in methodical terms of explanation generally found in methodologies. This lead to the decision of representing section data in such manner could be semantically is more useful for generic trait extraction for each section.