Data Mining, Management and Visualization in Large Scientific Corpuses

Hui Wei, Shaopeng Wu, Youbing Zhao, Zhikun Deng, Nikolaos Ersotelos, Farzad Parvinzamir, Baoquan Liu, Enjie Liu, and Feng Dong

Department of Computer Science and Technology, University of Bedfordshire, Luton, UK
{hui.wei, shaopeng.wu, youbing.zhao, zhikun.deng, nikolaos.ersotelos, farzad.parvinzamir, baoquan.liu, enjie.liu, feng.dong}@beds.ac.uk

Abstract. Organizing scientific papers helps efficiently derive meaningful insights of the published scientific resources, enables researchers grasp rapid technological change and hence assists new scientific discovery. In this paper, we experiment text mining and data management of scientific publications for collecting and presenting useful information to support research. For efficient data management and fast information retrieval, four data storages are employed: a semantic repository, an index and search repository, a document repository and a graph repository, taking full advantage of their features and strength. The results show that the combination of these four repositories can effectively store and index the publication data with reliability and efficiency and hence supply meaningful information to support scientific research.

Keywords: data management, distributed storage, NoSql, text mining, visualization, document repository, Elasticsearch, graph database.

1 Introduction

Digital libraries of scientific published papers are the main channel for acquiring cutting edge knowledge for scientific workers. This system presented is designed to help researchers overcome the limitations in pursuing scientific discovery through the analysis of published research papers. A large number of papers in computer graphics have been collected to study the trends of topic changes with the time. Most digital libraries concentrate on single document analysis. Our project DrInventor [1] tries to help the user overcome the difficulties in pursuing scientific discovery. The initiative is todiscover scientific creativity and technological change to facilitate scientific research. Researchers normally search citations from public digital libraries. They use keywords to find relevant publications from their titles, abstracts, mesh terms or full text. Due to the size of the libraries, finding the right papers is not easy and our system is designed to help them.

The remainder of the paper is organized as follows. In section 2 we describe how we collect, analyse and extract information from raw resources. In section 3 we discuss managing these data in suitable data repositories to help efficient querying related data. And in section 4, we present two visualization demos to display our processing results in two cases.

2 Data Collection

As these citations are semi-structured, they follow a certain template (e.g. ACM). The basic information we need has to be extracted from the original documents in pdf format. With the aid of further text analysis, we convert the original pdf files into plain text files by using Apache PDFBox [2], an open source Java tool for working with PDF documents. These original pdf files are kept since they are still useful to users. We separate these papers according to their publication data (e.g. year, volume). A corpus is defined with properties such as organization, conference, year. In each phrase we process one corpus. If there is a failure in one corpus, it will not affect the other corpuses.

The data collection phrase has three targets:

- extraction of basic information of a paper such as authors, title, abstract sentences, doi, which can be used to provide brief information about a paper and to provide a unique ID for each paper, hence avoiding duplicated processing.

- extraction of references, citation information, and hence from each paper to build up relation chains among the publications.

- extraction of standard keywords and their frequency from each paper. With these results, we could further calculate the topics from the information.

The data collection work is also used in EC project CARRE [3].

2.1 Metadata Extraction

We implement Metadata extraction by using a text processing pipeline supported by the GATE Text Engineering Framework [4]. A text file is broken down into sentences, then tokenized and POS tagged with the ANNIE system [5], followed by recognizing person names and numbers with gazetteers. With all this information, we then use a series of grammar rules (JAPE: Java Annotation Pattern Engine [6] rules) for the extraction of meaningful information:

1. Define “Macro”s from ACM format to find important markers, such as “ACM Reference Format” at the start of a converted txt file; “Abstract” tag, “Keywords” tag, “Introduction” tag, “DOI” tag , “year” tag, “Author” tag, “CR Categories” tag and so on. Make JAPE rules to output “author”, “year”, “title” tags.

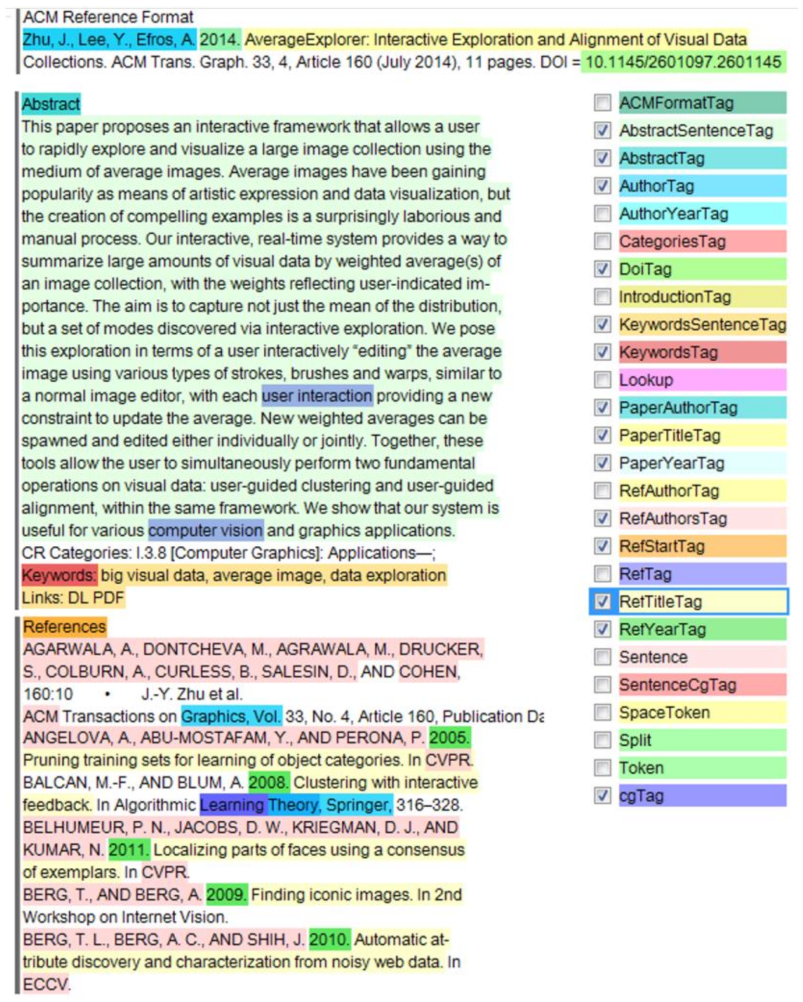
2. Search for two consecutive sentences with first sentence containing {authorTag}{yearTag} or {authorsTag}{yearTag}. From these two sentences, reference title, year, authors are extracted.

3. Search for abstract sentences, located between the "Abstract" tag and the "CR Categories" tag.

4. Search for keywords list sentences starting from "the Keywords" tag, ending at the "Introduction" tag.

The above rules apply to ACM publications, with some modifications needed over different time periods. By applying these rules, each paper is mapped to one metadata and is stored in the document repository.

Fig. 1. Shows the basic metadata information, including titleTag, authorTag, yearTag doiTag etc, extracted with multi tags. The citing list of one citation is extracted with citing title, author and year tags. Keywords information in cgTag is also extracted in context to help calculating keywords frequency.



ACM Reference Format
Zhu, J., Lee, Y., Efros, A. 2014. AverageExplorer: Interactive Exploration and Alignment of Visual Data Collections. ACM Trans. Graph. 33, 4, Article 160 (July 2014), 11 pages. DOI = 10.1145/2601097.2601145

Abstract
This paper proposes an interactive framework that allows a user to rapidly explore and visualize a large image collection using the medium of average images. Average images have been gaining popularity as means of artistic expression and data visualization, but the creation of compelling examples is a surprisingly laborious and manual process. Our interactive, real-time system provides a way to summarize large amounts of visual data by weighted average(s) of an image collection, with the weights reflecting user-indicated im-portance. The aim is to capture not just the mean of the distribution, but a set of modes discovered via interactive exploration. We pose this exploration in terms of a user interactively "editing" the average image using various types of strokes, brushes and warps, similar to a normal image editor, with each user interaction providing a new constraint to update the average. New weighted averages can be spawned and edited either individually or jointly. Together, these tools allow the user to simultaneously perform two fundamental operations on visual data: user-guided clustering and user-guided alignment, within the same framework. We show that our system is useful for various computer vision and graphics applications.
CR Categories: I.3.8 [Computer Graphics]: Applications—;
Keywords: big visual data, average image, data exploration
Links: DL PDF

References
AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN,
160:10 • J.-Y. Zhu et al.
ACM Transactions on Graphics, Vol. 33, No. 4, Article 160, Publication Da
ANGELOVA, A., ABU-MOSTAFAM, Y., AND PERONA, P. 2005. Pruning training sets for learning of object categories. In CVPR.
BALCAN, M.-F., AND BLUM, A. 2008. Clustering with interactive feedback. In Algorithmic Learning Theory, Springer, 316–328.
BELHUMEUR, P. N., JACOBS, D. W., KRIEGMAN, D. J., AND KUMAR, N. 2011. Localizing parts of faces using a consensus of exemplars. In CVPR.
BERG, T., AND BERG, A. 2009. Finding iconic images. In 2nd Workshop on Internet Vision.
BERG, T. L., BERG, A. C., AND SHIH, J. 2010. Automatic at-tribute discovery and characterization from noisy web data. In ECCV.

☐ ACMFormatTag
☑ AbstractSentenceTag
☑ AbstractTag
☑ AuthorTag
☐ AuthorYearTag
☐ CategoriesTag
☑ DoiTag
☐ IntroductionTag
☑ KeywordsSentenceTag
☑ KeywordsTag
☐ Lookup
☑ PaperAuthorTag
☑ PaperTitleTag
☑ PaperYearTag
☐ RefAuthorTag
☑ RefAuthorsTag
☑ RefStartTag
☐ RefTag
☑ RefTitleTag
☑ RefYearTag
☐ Sentence
☐ SentenceCgTag
☐ SpaceToken
☐ Split
☐ Token
☑ cgTag

Fig. 1. Metadata Extraction.

2.2 Keywords Extraction

The Microsoft Academic Search (MAS) API [7] allows developers to build applications by leveraging the data and functions of MAS. They supply a keyword function that represents keyword objects in many fields such as "Biology, Chemistry, Engineering Mathematics, Physics, Computer Science" and so on. In the computer science category, domains like "Algorithms & Theory, Artificial Intelligence, Computer Vision, Data Mining, Databases, Graphics" are described separately. We target our research in the "Computer>Graphics" domain, from which we collected

13,670 keywords. In a scientific publication, the usage of a group of keywords reflects its topic. We use these 13K keywords as standard terms to match phrases used in a computer graphics paper. These 13K computer graphics keywords are used for every citation to collect keyword frequency information. Then we need to store it in a suitable repository that can be easily mapped with standard words in the Computer Graphics (CG) citation context.

The keywords we fetched from MAS API contain a variety of terms for the same meaning, such as “three dimensional”, “3D” and “three-dimensional”. For machine processing, these terms will be treated as different terms. To understand these terms, an ontology is introduced into the system to share this conceptualization [8-9]. These “3D” synonyms should be treated as same “type” in the ontology with multi “same As” links. We defined predicate “rdf:type” and “owl:sameAs” for this purpose in Sesame [10] RDF repository and convert all these standard terms into Resource Description Framework (RDF) triples. By building up this “OWLIM-LITE” repository with “Owl-max” ruleset, this repository is connected as the ontology from GATE Gazetteer.

3 Data Repository

Apart from keywords stored in the RDF repository, a document repository CouchDB [11] is employed for metadata and a raw data repository. In order to enhance full text search ability, the search engine Elasticsearch [12] is used to store the abstract part of the metadata in JSON format. Graph databases are very efficient in traversing relationships [13], which are suitable for dealing with citing relations between citations. We benefit from these four repositories in terms of flexibility, efficiency, convenience, and intuition.

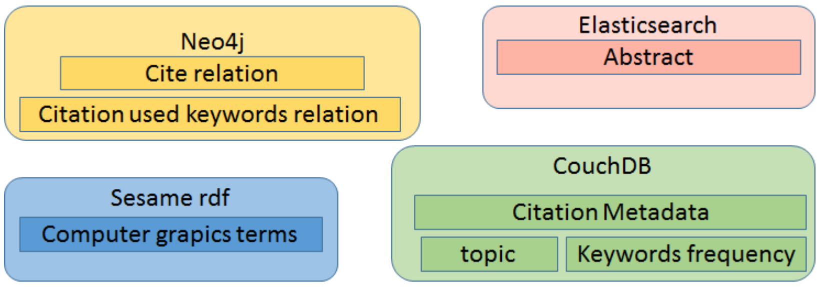


Fig. 2. Data is managed in four NoSql repositories.

We use Sesame RDF to store computer graphics (CG) terms or keywords as ontologies (fig. 2); the benefit of RDF storage is that we can update it when new terms are created or synonyms of an existing term are found. With inferencing and querying support, our ontology is connected with GATE as language resources. Then this language resource is used to create an ontology Gazetteer in GATE. To recognize these CG terms, a Jape Transducer with JAPE rule file is defined to pick out these terms and give them a unified tag (cgTag in fig. 1).

The metadata that describes the basic information of a citation is managed in the Document repository, CouchDB. Keyword frequency of a document reveals what a citation talks about with standard domain terms. This information is stored in CouchDB at the data collection stage. Abstract information from a citation is an important summary. This data extracted from the raw PDF file is stored as an attachment at the data collection stage.

CouchDB is designed for web application, a raw PDF citation file can be stored as an attachment to its metadata. The attachment presented by a URL can be easily passed to another file consumer, since a URL mapped to this file can be identified by its document ID. As a schema-free database, one document's data is allowed to have any number of properties. From that character, inheritance classes could be mapped into the same data document. That means it gives the user the freedom of mapping data with more properties or fewer. Views in CouchDB is a kind of design document which supplies output of a group of related data. As these view design documents are written by the user through JavaScript, the calculation ability is more flexible than SQL. Validation function is a special property of a design document. From both old and new document content as input, this function gives the user the ability to verify relation of properties or relation of new and old documents. The Map/Reduce function supplied by CouchDB is useful for aggregation data to create a summary of a group of data.

Although views are the tool for querying data on CouchDB documents, full text search cannot be achieved even through Elasticsearch Rivers [14]. Elasticsearch is introduced for this propose. At the data processing stage, citation data (abstract information, a few basic metadata properties and document ID) read from CouchDB is loaded into the search engine. As a shared entity, the citation object is stored in the Document repository with the full set of properties, while in search engine the citation object with basic properties is loaded. The link between these two cross-repository objects of the same citation is the same ID. So the citations found by the search engine can be tracked with full information in the Document repository.

Graph databases are very efficient in traversing relationships. We select Neo4j as the Graph database storing relationships of citations and relationships between citation and keywords. Relationships between citations can be defined in many cases. The citing relationship describes the reference list of a citation. This information can be gathered from the citation itself. The cited relationship describes other citations which use this given citation as a reference. This information also can be gathered from the Document repository by using a query. But detecting similar citing or cited papers between two given papers needs massive deep querying that is not efficient. So the Graph database is used in our system. It loads data from the Document repository to create relationships of reference (citation) and usage (keywords) at the data processing stage. In the graph repository, a relation can be described by a path; it is more efficient and convenient to find a path in the graph repository.

4 Citation Data Visualization

4.1 Cited Relationship

As mentioned before, reference information is extracted from the raw PDF file. Due to formatting problems, some references are extracted without exactly matching a paper title even though they exist in the Document repository. But we still have a large number of relations in the Graph repository. At this stage we treat citations within“SIGGRAPH” as internal papers, citations from other digital libraries like IEEE as external reference papers. As we have processed metadata from raw SIGGRAPH for 13 years of citations, one SIGGRAPH paper citing other SIGGRAPH papers can be tracked, otherwise these cited papers can only be treated as references.

In order to extract meaningful data, we present our “be cited” relationships in a D3 [15] chord diagram.

The data structure in the Graph repository can be treated as many trees if we treat starting nodes as the root of its reference tree. In our Chord diagram, only root nodes and leaf nodes of the required length path are presented.

Fig. 3 (F) presents 13-year paper citing relations. The data are papers that have relationships between them more than length 6 in 13 years. Short arc papers have normally been published in recent years and the longer arc papers normally in earlier years. Querying these inheritance relations from the Graph database is much more efficient than in other types of database. The longer the arc, the more length 6 citing papers are related to the paper. These longer arc papers are more likely to be root nodes of depth 7 tree structures. Fig. 3 (G) presents the relation between the given single citation and its indirect citing papers with mouse hover on a paper title. The diagram shows only citing papers of this input paper and hides other papers. Due to image space limitations, all paper titles are reduced to 20 characters. Fig. 3 (H) presents citations as cited citations in the year 2013 corpus and their citing citations. As we have 13 year corpuses from 2002 to 2014, those citing papers are published papers from the year 2014 corpus. From H we can easily recognize a paper in brown named “Globally optimal dir” is cited more times than others.

Topic Distribution

Fig. 4 uses keywords data linked with the paper year property to calculate topic modelling. The probabilities of topics are shown according to their proportion. The topics were retrieved by the Latent Dirichlet Allocation (LDA) algorithm. The algorithm assumes the documents were produced in a probabilistic generative model, which discovers the topics in every document or a corpus. The figure shows six selected topic variations from year 2005 to 2014. The proportion of the topic is represented by height at each time along the horizontal axis. The topic colours correspond to the legend colours and texts. The interactive functions would offer more information, e.g. mouse hover on a topic of certain year. Then the topic name, year and value of the topic are shown with the pointer. Comparing to [16], which uses a matrix-like style to present topics, ours has more explicit and simple visibility and operations.

The project only concentrates on the computer graphic documents from proceedings of ACM SIGGRAPH. For example, a decreasing publication number can be seen in the topic of “image compression” while a chopping pattern shows in the topic of “real time”. From these trends, the users can judge research topic trends by years. More information, such as topic contents, topic-related documents and their publication details, will be added in the next stage.

5 Conclusion

In this paper, we present our work on text mining and data management on a large number of scientific publications for collecting and presenting citation information and topic trends to facilitate research work. Four data storages including a semantic repository, an index and search repository, a document repository and a graph repository are employed for efficient data management and fast information retrieval. The experiment results show that the combination of these four repositories can efficiently store and index the publication data reliably to supply valuable information to support scientific research, which further helps researchers to derive meaningful insights of the published scientific resources more conveniently, enabling them to grasp technological change more quickly and hence assists new scientific discovery.

Acknowledgments

The research is supported by Dr Inventor project {the European Union Seventh Framework Programme ([FP7/2007-2013]) Dr Inventor under grant agreement no. 611383} and CARRE project {the Seventh Framework Programme of European Commission – ICT under agreement of FP7-ICT-611140}.

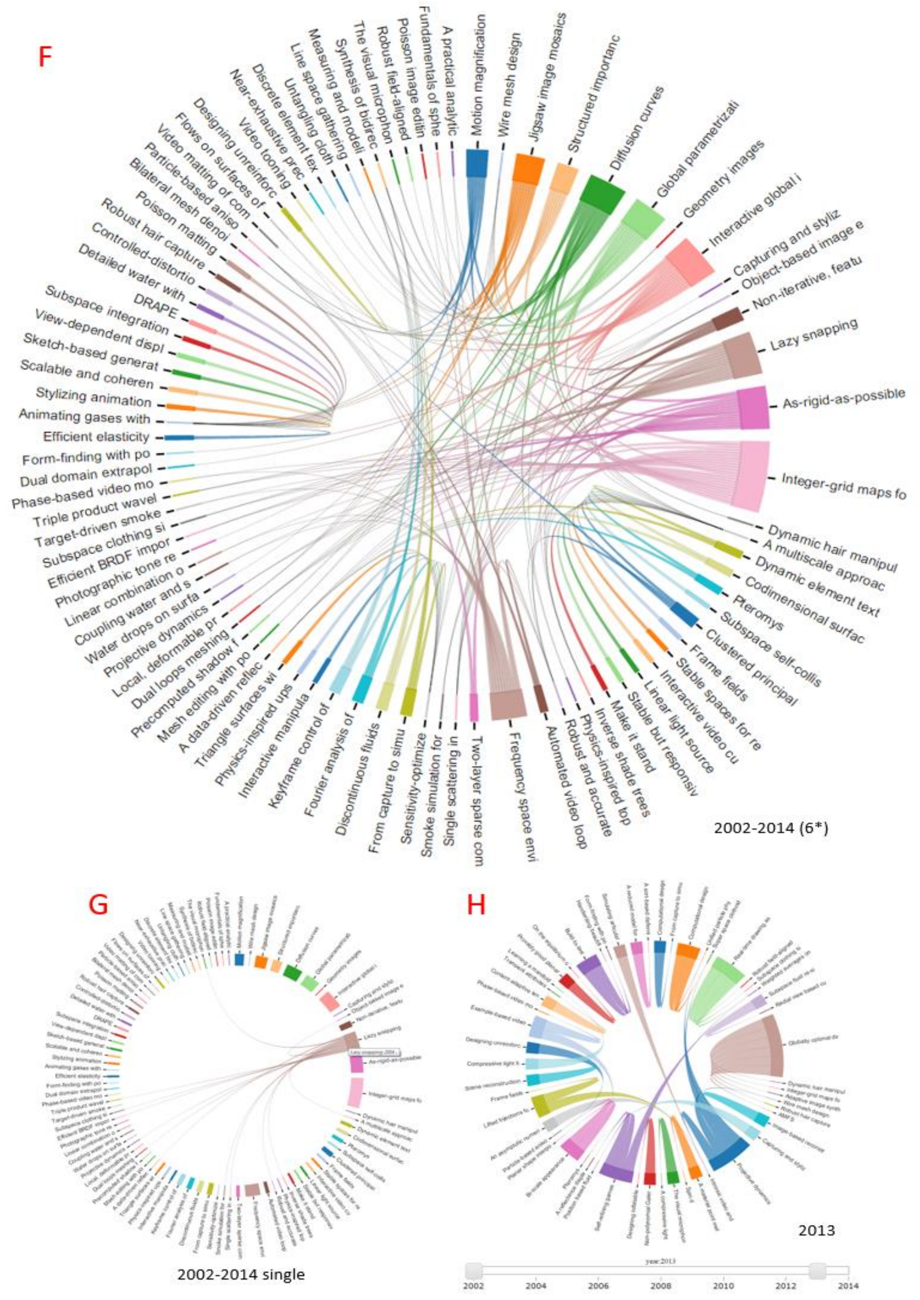


Fig. 3. Chord diagram. F: 13-year corpuses of length 6 cited relations. G: singlecited relations. H: one year cited relations.

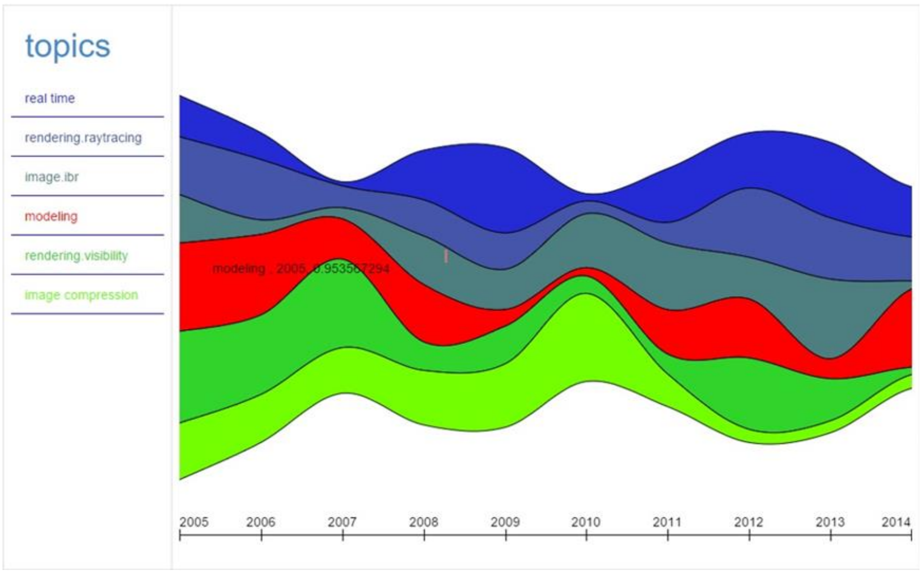


Fig. 4. Topic river visualisation

References

1. DrInventor, http://drinventor.eu/
2. pdfbox, https://pdfbox.apache.org/
3. CARRE, https://www.carre-project.eu/
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan., V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002
5. ANNIE https://gate.ac.uk/sale/tao/splitch6.html#chap:annie
6. Thakker, D., Sman, T., Lakin, P.: GATE Jape Grammar Tutorial", Version 1.0, A, Pictures, UK. (2009)
7. Microsoft Academic Search (MAS) API http://academic.research.microsoft.com/
8. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquistition, 5 (2) 199-220
9. Jin, L., Liu, L.: An Ontology Definition Metamodel based Ripple-Effect Analysis Method for Ontology Evolution, in Procs. 10th International Conference on Computer Supported Cooperative Work in Design, 1-6, DOI:10.1109/CSCWD.2006.253032
10. Fensel, D., Hendler, J., Lieberman, H., Wahlster, W., Berners-Lee, T.: Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. in MIT Press eBook Chapters: Spinning the Semantic Web:Bringing the World Wide Web to Its Full Potential, (2005) 197–222

11. CouchDB http://couchdb.apache.org/
12. Elasticsearch https://www.elastic.co/products/elasticsearch
13. Grolinger, K., Higashino, W.A., Tiwari, A.,Capretz, M.A.M.: Data management in cloud environments: NoSQL and NewSQL data stores. Journal of Cloud Computing: Advances, Systems and Applications, 2 (22) (2013) 2-22, doi:10.1186/2192-113X-
14. Elasticsearch Rivers https://www.elastic.co/guide/en/elasticsearch/rivers/1.4/index.html
15. D3, http://d3js.org/
16. Alexander, E., Kohlmann, J., Valenza, R., Witmore, M., Gleicher Serendip, M.: Topic model-driven visual exploration of text corpora, Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on, 173 -182