Crowd-annotation and LoD-based semantic indexing of content in multi-disciplinary web repositories to improve search results

Arshad Khan
School of Electronics & Computer Sciences (ECS),
Building 32, University Road,
Highfield, University of Southampton
+44 (23) 59 7750
a.khan@soton.ac.uk

Thanassis Tiropanis
School of Electronics & Computer Sciences (ECS)
Building 32, University Road,
Highfield, University of Southampton
+44 (23) 59 9109
t.tiropanis@soton.ac.uk

David Martin
National Centre for Research Methods (NCRM)
Building 44, University Road,
Highfield, University of Southampton
+44 (23) 59 3808
d.j.martin@soton.ac.uk

ABSTRACT

Searching for relevant information in multi-disciplinary web repositories is becoming a topic of increasing interest among the computer science research community. To date, methods and techniques to extract useful and relevant information from online repositories of research data have largely been based on static full text indexing which entails a 'produce once and use forever' kind of strategy. That strategy is fast becoming insufficient due to increasing data volume, concept obsolescence, and complexity and heterogeneity of content types in web repositories. We propose that by automatic semantic annotation of content in web repositories (using Linked Open Data or LoD sources) without using domain-specific ontologies, we can sustain the performance of searching by retrieving highly relevant search results. Secondly, we claim that by expert crowd-annotation of content on top of automatic semantic annotation, we can enrich the semantic index over time to augment the contextual value of content in web repositories so that they remain findable despite changes in language, terminology and scientific concepts. We deployed a custom-built annotation, indexing and searching environment in a web repository website that has been used by expert annotators to annotate webpages using free text and vocabulary terms. We present our findings based on the annotation and tagging data on top of LoD-based annotations and the overall *modus operandi.* We also analyze and demonstrate that by adding expert annotations to the existing semantic index, we can improve the relationship between query and documents using Cosine Similarity Measures (CSM).

Keywords

Crowd-annotation; semantic search; Linked open Data; semantic annotations; tagging and annotation; web repositories search; Elasticsearch;

1. INTRODUCTION

The primary goal of any searching or retrieval system is to structure information so that it is useful for people in finding desired and relevant information effectively and efficiently.

Current searching techniques in discipline-specific or multi-disciplinary repositories[1] predominantly use keyword instances in web documents where users rely on the incidental mention of keywords and phrases. Contemporary research users struggle to filter out irrelevant information especially in a scientific discipline where relevance and precision are of great importance to support ongoing research studies. Another aspect of this issue can be highlighted through the lens of time, which changes the meanings of various concepts, terminologies and things on the Web thus making it difficult for search engines to serve online users and the research community using the same Boolean search model.

However, search engines have experienced impressive enhancements in the last decade, but information searching is still keywords-based which falls short of meeting users' needs due to insufficient content meaning [1]. Similarly [2] describes the basic Web search as inadequate when it comes to finding contextually relevant information in web archives or collection of websites like ReStore[2] repository. We have been using this web repository website (currently used by 15000 plus user /month) as a test bed for LoD-based semantic and crowd-annotation. We have also used this website for the deployment of the Elasticsearch[3] semantic search application in the past and published our findings in [3].

Current search engines are no more able to really help the user in tasks that go under the umbrella of exploratory search. Here,

ACSW '17, January 31-February 03, 2017, Geelong, Australia
© 2017 ACM. ISBN 978-1-4503-4768-6/17/01…$15.00
DOI: http://dx.doi.org/10.1145/3014812.3014867

[1] A web repository stores and provides long term online access to a collection of web sites or web resources (containing static & dynamic web pages), research papers, presentations, experimental code scripts, reports etc. funded by UK research councils. Examples include http://www.dataarchive.ac.uk/,http://www.timescapes.leeds.ac.uk/

[2] ReStore is an online repository of web resources developed as part of Economic & Social Research (ESRC) council funding-available at http://www.restore.ac.uk.

[3] Elasticsearch is a flexible and powerful open source, distributed, real-time search and analytics engine. Available at http://www.elasticsearch.org.

the user needs not only to perform a look up operation but also to discover, understand and learn novel contents on complex topics while searching [4]. The inability to designate unambiguously the rapidly growing number of new concepts generated by the growth of knowledge and research in a scientific discipline such as social sciences [5] is another issue failing the traditional search engines. Such issues have partly been addressed by keywords based searching where plain keyword queries are converted into equivalent semantic queries followed by syntactic normalization, word sense disambiguation [6] and noise reduction. To do that, the use of dictionaries (e.g. Wordnet), thesauri and other library classification systems have been exploited in collaboration with the domain specific ontology to express keywords in more structural language. The semantic keywords are then matched with ontology terms and various semantic agents are applied to disambiguate terms before retrieving the results [7]. All such approaches tend to distort the users' actual queries [8] thus causing ambiguous queries to lead to less relevant and imprecise search results.

However, as described above, like other information domains, in scientific research disciplines terms change overtime due to cultural, social, technological, scientific and socio-economic etc. factors which compromise relevance and accuracy in search results. All this suggests that semantic expressions and matching terms with ontologies classes/properties (linguistic) and instance data (semantic information) will not be long-lived and would need frequent and regular expert human intervention.

To further investigate the above-mentioned issues we have been focusing on 2 main areas as part of this research. (a). Whether obsolescence in terms and concepts in online repositories of social science could be addressed by incorporating in-page annotation environment (as opposed to Social bookmarking based tagging [9]) and real-time modification of semantic index with authentic annotation and tags. (b) Whether document and query relevance could be improved by using a Semantic Vector Space (SVS) model, where search results retrieval takes into account semantic entities, concepts and crowd annotations in ranking the top 10 results in a typical search application.

We have presented web resources development and archival process extensively in [3] which delineates the entire process flow involving UK research funding councils, multi-disciplinary teams of researchers, higher education institutions and publication of research outputs in institutionally funded websites or repositories. This paper is an extension of that work with a specific focus on expert crowd-sourced annotation in web repositories, and ranked retrieval of information using Elasticsearch distributed search application. We have also worked alongside academic social scientists and library sciences professionals to upgrade a classification system called the NCRM Typology, which has been extensively used, in the classification of social science research outputs in the UK. The NCRM Typology classification was completed after six months' review in January 2015[4]. We have thoroughly deployed the typology in our annotation, indexing and searching framework to assess the effectiveness of vocabulary-based annotation and tagging vis-à-vis free-text tagging.

The remainder of this paper is organized as follows. In Section 2, we will review relevant work carried out in this area. In Section 3, we will outline our methodology and the entire process flow of indexing crowd-annotation and tagging in the ReStore web repository on top of the semantic annotation layer. Section 4 will explain query formation and search results retrieval from our Elasticsearch semantic search engine. Section 5 will detail the improved document and query relationship in an SVS model leading to changing document ranks following crowd-annotation and tagging of webpages by expert annotators. Section 6 will describe future work and conclusions.

[4] http://eprints.ncrm.ac.uk/3721/

2. RELATED WORK

A substantial amount of research has been conducted in this area where the emphasis has been, for example, on ontology-based information retrieval [10], query expansion-based searching[1, 11], social annotations based on social bookmarking platforms [12] and key-phrase extraction based on semantic blocks [13]. We understand that designing and evolving domain-specific ontologies still remains a challenge, in the face of ever expanding web repositories where scalability and content heterogeneity are of great importance. The level of complexity and time taken to refine ontological classes and their relationships with external sources of data (e.g. concept disambiguation, word sense and term stemming) are challenges to scalability. The problem is further complicated when addressed in a multidisciplinary research environment where experts are scarce and unlikely to be motivated to take part in the evolution of a domain-specific ontology framework. Our own prior work [14] is testimony to this challenge where the support could not be obtained from the broad research community but population of a small corpus of static documents in a client-server architecture was too constrained to be scalable. Furthermore, establishing ontology as a semantic backbone for a large number of distributed web resources is not easy, as different actors will have different views on what exists in these web resources. This all implies that carving out a general purpose ontology, fitting all resources [9] is almost impossible. The fact that human intelligence is more accurate [15] when it comes to interpreting text in documents or web pages further limit the role of a general purpose ontology for semantic annotation. Bontcheva, Tablan [15] further highlights the challenges of retrospective and prospective human annotation to justify the role of general purpose ontology but we understand that the human role is inevitable due to the complex nature of content and their volume when it comes to search and retrieval in scientific repositories.

Another major problem the semantic web community faces for the construction of innovative and knowledge-based web applications is to reduce the programming effort while keeping the web searching task as small as possible [16]. Several studies have been conducted to explore social annotation as one of the enabler platforms for implementing semantic annotation and information retrieval. However, the fact that tags are chosen by the user without conforming to a priori dictionary, vocabulary ontology or taxonomy [9], it's hardly adopted for an in-house multidisciplinary search application. The obvious problem with social annotations is that they are made by a large number of ordinary web users without reference to a pre-defined ontology or classification system such as in the case of Delicious [9]. Social bookmark services no doubt provide a pragmatic user interface for users to annotate content, but the challenge remains that without clear semantics, social annotations won't be of much use for web agents and applications on the Semantic Web [9]. All this means that web search platforms built on top of social annotation-based platforms, are unlikely to yield relevant search results in the face of the ever-increasing web of information.

Social semantics is another area of research, defined by the interaction and socialization of users along with user-generated content, which in most cases does not conform to a classification system. The tacit agreement on their usage and understanding, however, make social semantics an important element of web search but they stand in contrast to the more logical semantic web [17]. Another issue, arising from our experience of setting up a fully-fledged annotation and indexing platform, is that Social bookmark service providers enable tagging on webpages, but analyzing tags (e.g. free from stop words, redundancy and ambiguity) and mapping them on to the relevant record in an inverted index still remains an issue. The level of noise in the resulting tag clouds does not usually produce a meaningful or semantically related tag cloud.

The DBPedia-based approach to tagging and information retrieval by Mirizzia, Di Noiaa [4] is impressive but the overreliance on Wikipedia and the fact that all tag suggestion have to come from Wikipedia's labels, categories and abstracts makes their approach somewhat restrictive. The fact that every tag suggestion has to come via a RESTful endpoint from DBPedia only and not from a domain-specific tagging environment makes it potentially ineffective in a scientific web repository. Quality determination of the LoD dataset is another issue highlighted by [18] which matters a lot in a scientific web repository, having heterogeneous types of content, in order to develop new web-based services for knowledge discovery and data exploration.

Social annotations are emergent useful information that have been used as part of web search in terms of folksonomy, visualization and semantic web [12] but to our knowledge annotations and vocabulary-based tags have not been used as part of a full-fledged semantic indexing and searching environment.

Page ranking-based tagging is another issue [12] has been discussed, implying that a page only becomes annotable or "taggable" when it has achieved a certain level of page rank popularity. In that situation, another framework is required which pre-populates tags based on semantic, lexical and crowd-annotations in an auto-complete type of environment so that the granularity of tag suggestions could be increased from a 1-2 words, Wikipedia-based labels suggestion to a more diverse suggestive system.

Another issue worth highlighting, and which many semantic search systems suffer from, is the usability of such systems at the time of seeking inputs from users i.e. annotating content and/or specifying query for searching. Users are expected to use formal query language to express their requirements, which is not usually the case in online search applications. A lack of optimal semantic annotation of content in web documents based on a small set of pre-defined domain ontologies and datasets [1] further limits the overall purpose of tagging and annotating.

3. OUR METHODOLOGY

We propose that by incorporating dynamic Linked Open Data (LoD) based semantic indexing, enhanced by crowd-sourced annotations, and vocabulary-based tagging (NCRM Typology), we can address the issues of content heterogeneity, volume of data and terminological obsolescence in repositories of web resources in a typical research domain of social sciences. A vocabulary or typology in this case, in any scientific field is a collection of terms, concepts and terminologies, which contemporary researchers use to refer to various things in that field. Multidisciplinary web repositories contain data or research outputs that are produced by researchers within disciplinary domains (e.g. social sciences). We base our analysis on augmenting the existing content metadata utilizing automatic LoD based semantic annotation and indexing followed by crowd-annotation (free text and vocabulary annotations techniques). We have used the Restore (www.restore.ac.uk) repository for all the annotation and tagging experiments discussed in this paper.

Our experiments and analysis will show (a) whether crowd-sourced annotation can be effectively used for better information retrieval (b) whether semantic indexing and retrieval of knowledge can be enhanced after new terms and concepts are introduced through a domain-specific concept vocabulary (and applied by the crowd-sourced expert annotators) and (c) how best to represent a natural language query in terms of semantic query to enhance its contextual similarity with documents in a semantic search environment.

In our framework, firstly, control over creating and tuning the tokenizers and analyzers addresses the issue of disambiguation and redundancy at the outset, before the documents are even indexed. Secondly, the fact that we can map vocabulary keywords to semantically related keywords in the form of synonyms, gives us further control at the time of filtering search results at the time of searching. Thirdly, the availability of free-text popularity-based tags and vocabulary-based tags make the task of semantically relating various content far more trustworthy and sustainable in the face of fast-changing user terminologies and scientific concepts. We will discuss this further in the forthcoming sections.

3.1 Semantic Indexing

We have automatically indexed 3400 documents, which include html, shtml, PHP, word, pdf files. We have extracted topical keywords, concepts, and entities along with relevance scores and sentimental values from all these documents and stored them in a dedicated index in our Elasticsearch cluster. At this stage, we have already amalgamated the inverted index with the semantic index and thus we call it a semantic index to distinguish it from full-text index only. We have used Alchemy API[5] due to its holistic approach towards text analysis and broad-based training set (250 times larger than Wikipedia) used to model a domain like ours. The Alchemy API platform uses Machine Learning (ML) and Natural Language parsing algorithms for analyzing web or text-based content for named entity extraction, sense tagging and relationship identification [19]. The platform was also one of the best in the performance evaluation review of [20]. Alchemy API remained the primary option for NE recognition, overall precision, recall of NEs, types inferences and URI disambiguation. All documents are passed on to the Alchemy APIs i.e. text analyzer to extract topical keywords, concepts and entities. The full text, title, size, date of indexing are stored alongside the semantic concepts, entities and keywords.

The index creation stage is important in its own right in that we have defined a fully-fledged schema for the index, which comprises of selecting appropriate nGram tokenizers at the

[5] AlchemyAPI provides RESTful API endpoints for all text-mining and content analysis functionality with a special privilege to us (academic research) for analysing 30, 000 URLs per day

parsing stage and standard analyzer (white space) at the searching stage. It also involves mapping domain specific terms, acronyms, concepts and jargon to equivalent synonyms that are then applied on each document at the time of indexing or preprocessing of documents. Standard tokenization of terms in documents means that each term in a document has been tokenized and represented in the document by words, stems or lemmas of words as well as character n-grams. So essentially Elasticsearch applies the search analyzer on all components of semantic index based on the tokenization defined at the time of scheme design to retrieve relevant information. The components of the semantic index include full text, semantic representation of natural language text i.e. keywords, concepts and entities and crowd-sourced annotations which further comprises of both free-text and vocabulary annotation or NCRM Typology-based annotations.

Figure 1 shows the overall process flow of a document's journey from the repository website to Elasticseach index and storage via Alchemy API and then the modification of each document with annotation and tagging at a later stage. The map of the semantic index schema and a typical document indexing and storage in Elasticsearch has been demonstrated and available at http://goo.gl/QpFmpf.

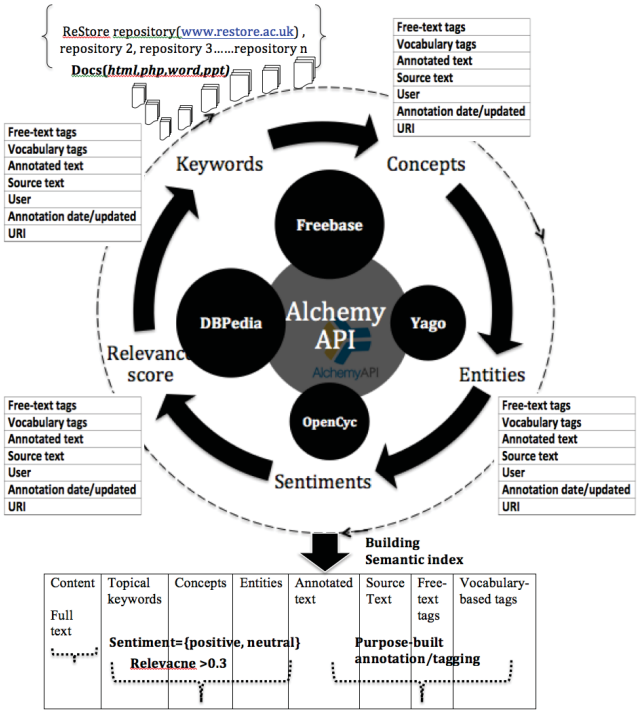

Figure 1: Semantic indexing, crowd-annotation of semantic index with semantic relevance and TF-IDF scores

3.1.1 Mapping synonyms to typology categories

We have also experimented with stuffing synonyms into the schema of semantic indexes at the time of indexing documents. The purpose of mapping synonyms to obsolete natural language terms has to address the issue of concepts obsolescence (partially if not entirely) in a scientific discipline. Obsolescence of concepts and meanings occurs in such disciplines over time and they need to be replaced with contemporary concepts and terms from time to time to ensure the searchability of content in a scientific repository of web content. We have mapped significant classes in the NCRM Typology to potential terms that could be found in the entire document corpus, which as per our claim will improve similarity between documents at the time of retrieval.

3.2 Elasticsearch (ES) as a Knowledge Management Platform

Elasticsearch[6] (ES) is a flexible and powerful open source, distributed, real-time search and analytics engine. We have deployed Elasticsearch server on a shared ReStore repository server platform used as the back-end of our annotation and tagging environment as well as in its capacity as a search application. The size of the current hybrid semantic index) full-text, LoD-based annotations and crowd-annotations) is 240MB with 8GB of physical memory for sharing with the ReStore web server. Two analyzers for indexing (*nGram*) and searching (*Whitespace*) have been created alongside *stopwords* and *synonyms filter* to map domain and scientific-discipline-specific acronyms (89 in total) to actual text and cut the size of document vectors from the outset. Synonyms also act as the best linkage source between web documents at the time of retrieval. We have used the Elastica[7] library to embed the annotation and tagging tool into the ReStore repository website to facilitate an intuitive user interface to human annotators. We have also used it for rendering a complete web-based search application to analyze search results based on automatic semantic annotation as well as crowd-annotation.

The ES scoring algorithm is a combination of both Boolean model and VSM Information Retrieval models. All documents that pass the Boolean model then go on to scoring with the VSM. The basic or standard formula for score calculation (without manipulation) is given as follow:

$$\text{Score}(q,d) = \text{queryNorm}(q) * \text{coord}(q,d) * \sum\left(\text{tf}(t \text{ in } d) * \text{idf}(t)^2 * t.\text{getBoost}() * \text{norm}(t,d)\right) (t \text{ in } q) \quad (1)$$

Where score (*q,d*) is the relevance score of document d for *q*, *queryNorm (q)* is the query normalization factor, *coord(q,d)* is the coordination factor, the sum of the weights for each term *t* in query q for document *d is tf.idf , t.getBoost()* is the boost factor applied to the query and *norm (t,d) is the field-length norm* which implies that that the shorter the field length, the greater will be the weight of the term in it. The above equation (1) is the modified version of this equation, which is given as the score for a document given a query

$$\text{Score}(q,d) = \sum_{t \in q \cap d}^{N} tf.idf\ idf_{t,d} \quad (2)$$

We will elaborate computation of log-weighted TF-IDF and subsequently carry out comparison of various vectors with query in Section 5. Elasticsearch analyzers first analyze all the content belonging to each document via JSON-formatted URLs and relevant scores are stored against keywords, entities and concepts (extracted by Alchemy API) using three different API services i.e. Keywords, Entity and Concepts). Each document D_j represents a vector space model in the following manner: $D_j = (t_k, t_e, t_c \ldots, t_{kec})$

Where t_k, t_e, t_c, t_t are the keywords (k), entities(e) and concepts (c) terms. The document vector is then modified by the expert annotator after adding more contemporary scientific annotations and vocabulary tags so the modified vector the becomes: $D_j = (t_k,$

[6] Available at http://www.elasticsearchorg.

[7] A PHP client for search available at http://elastica.io/.

$t_{e,}$ $t_{c,}$ $t_{aT,}$ ….., t_{kecaT}). With such representation, each document vector then has the power of influencing ranking of search results.

3.3 Methodology of crowd-annotation and tagging: storage and retrieval

The semantic web has yet to reach widespread usage. Collaborative tagging systems are now part and parcel of most major websites and their users seem to be increasing rather than decreasing [17]. Furthermore [17] elaborates that there are concrete benefits to the tagging approach compared to the Semantic Web's traditional focus on formal ontologies. The flexibility of tagging systems is thought to be an asset, which is a categorization process as compared to pre-optimized classification process such as expert-generated taxonomies. It is also a fact that to sustain taxonomical or ontological classification, a number of experts are required to review the axiomatic classification of new terms and concepts and then to populate the documents corpora with it for Knowledge Base (KB) creation. In a tagging environment, however, users are enabled to order and share data more efficiently than using classification schemes, as associating free text with content in a webpage is cognitively simpler than decisions about finding and matching existing categories. [17].

However, in our annotation and tagging framework, we have fed popular tags as well as prominent vocabulary tags in the form of an autocomplete list which maps users' cognitive thinking at the time of assignment of annotations. We have actually observed 6 of the 27 annotators (explained in section 3.5) while annotating content and almost all made use of the autocomplete list as the list always kick-started the thinking process of assigning a keyword without giving a clue to the annotator whether the keyword was free-text or vocabulary-based (i.e. borrowed from the NCRM Typology.

Our approach amalgamates semantically interpreted concepts, entities and topical keywords using not only Wikipedia [21] but other established data sources as well i.e. Freebase, Yago, OpenCyc and GeoNames. The top layer of crowd-sourced annotation then super-imposes a contemporary tags and annotations layer in order to sustain relevance at a higher precision and low recall with the passage of time. This also has to do with the rarity element of IDF and the field length norm i.e. *norm (t,d)* in equation (1).

3.4 Experts' annotation and tagging

The aim of this research phase is to determine whether users' experience of searching in online research data repositories could be improved by providing means for collaborative annotation of webpages. There are two phases of this study i.e. (a) annotation and tagging of content by the research community in online repositories of multi-disciplinary research data and (b) exploiting annotation metadata obtained from (a) to improve searching in those repositories. We also want to identify and piece together semantically related resources (webpages) based on users' interests, number of users tagging particular web resources and the kind of tags (free text and vocabulary tags) they are using for various webpages they have annotated. We assume that webpages are semantically related if they are tagged by a number of users having similar research interests. We also infer from our experience while observing many participants annotating and tagging web resources, that related web resources are usually tagged more than once by semantically related tags such as team management, "group dynamics", team leader, "project management", "research team", "leadership", "research team leader", "people skills", "research data management", "data sharing", "dissemination", or professional development as type degree. We have setup a generic annotation page for demonstration purpose, which can be accessed at http://goo.gl/MEJIze.

3.5 Recruiting Participants

We recruited participants by displaying posters in academic Schools' foyers (Education, Social Sciences, Geography, Statistics, Psychology, Computer Sciences), writing directly to module leaders in the Faculty of Social & Human Sciences in Southampton and module leaders in Edinburgh, Cambridge, Cardiff, Manchester, Loughborough, Warwick, Kent, Portsmouth Universities in the UK to forward the posters to Post-doc researchers and PhD students in their respective departments. We also directly approached some research fellows, web resource authors in the ReStore repository and professionals via their connection with the University of Southampton e.g. UK Data Service[8], Language & Computation research group in University of Essex[9] and requested their participation. Despite the enormity and novelty of the task i.e. annotating text and tagging webpages using both free text and vocabulary annotations, we were still successful in getting sufficient participants who were both curious and motivated in participating in the study. This approach helped in filtering out unwanted annotation and tags from the outset. We also mass-emailed PhD students only at the School of Social Sciences at Southampton[10] seeking their participation with options to either participate in focus group annotation/tagging study or attempt independent annotation, following guidance materials sent out by emails. A webpage containing information about the study and joining details aimed at PhD students. was created on the ReStore website at http://www.restore.ac.uk/focusgroup.

3.6 The annotation/tagging experiments

We aimed from the very outset at post-doc and PhD researchers as participants of this study in order to set a high pitch and obtain a gold standard annotation and tag benchmark for search results ranking and Precision-Recall-based IR. Focus group sessions were initially conducted with local PhD students in order to assess the level of difficulty in understanding and attempting the task and then refining the grey areas pointed out during observations for the next focus group session. Some students however preferred to attempt the study at their own computers in which case, consents were obtained via email and guidance materials were sent out in separate emails. A total of 27 expert participants annotated 450 webpages with 640 comments-based annotations on content of webpages, and 1670 typology (or vocabulary) and free text tags. The typology-based tags comprise of two levels: a broader level called `vocabularyAnnotation.level_1.level1` and a narrower level called `vocabularyAnnotation.level_2.level2` in query formulation. Annotators made use of 17 different broader level typology tags in 298 instances while 66 different narrower level typology tags were used in tagging webpages 318 times. The *allinOne* field in the *tagging* slider plugin offers the *autocomplete* feature to annotators based on typology terms as

[8] https://www.ukdataservice.ac.uk

[9] http://lac.essex.ac.uk

[10] http://www.southampton.ac.uk/socsci

well popular tags (*used at least 3 times*). More than 400 typology broader and narrower terms are offered through the tagging slider annotation plugin (through autocomplete) to enable annotators to assign at least 3 different tags to each webpage being annotated. The autocomplete not only facilitates existing word selection but also influences new keyword formulation which leads to establishing new relationships between documents at the time of information retrieval. For example, "*comparative methodology*", "*aggregate data*", "*sociolinguistics*" from socio-demographic, *evaluative assertion analysis*, "*economy, society and space*" from economy, "*critical discourse analysis of text*" from discourse analysis, "*corpus linguistics*" from corpus & documentary analysis and so on. A demo page[11] has been setup for the sake of this paper to show the annotation tools in action. Annotators took 90 minutes on average to complete the task of annotating/tagging 15-20 webpages but they had the freedom of attempting it at their own convenience by logging on to the system. This approach was adopted to distribute the participants in two groups i.e. focus groups for local participants, to understand their behavior to annotation & tagging and improve the system on the fly, and those intending to complete at the place of their choice in multiple intervals of time.

3.7 Scope of annotation and tagging

The aim of annotation and tagging in our framework is to use purpose-built website-embedded annotation and tagging tools to perform annotations and tagging in webpages only. The annotation and tagging environment become available to participants using individually created credentials. A login page[12] is used to access to the entire ReStore website for annotation and tagging purposes. The annotable webpages include both static and dynamic webpages, which ensures even access to all webpages. We provided a list of pre-selected URLs to some expert annotators based on their research interests and their preference for certain social science topics. We also sent out 1000[13] pre-selected keywords harvested from Google Analytics which had been submitted by online users (15000 approx. per month) as part of full-text searching on the ReStore website. These keywords were intended to motivate participants to use meaningful queries in order to find webpages for potential annotations. However, an equal number of participants preferred to use the online full-text search application to find webpages based on their research topics for annotation and tagging.

3.8 Questionnaire and participants feedback

To provide for the basic usability components i.e. learnability, efficiency, memorability and satisfaction [22] and measure them in each case, we asked for participants' feedback at the end of each individual annotation exercise. The short questionnaire included questions such as how desirable was it to annotate a piece of text or tag an entire webpage, the usability of both annotation and tagging tools, the suitability of typology terms for associating with webpages, and their willingness to assign their own keywords for tagging a set of webpages. We have used a 5-point Likert scale for expressing participants' feedback i.e. "*Strongly Agree", "Agree", "Neither agree nor disagree", "Disagree"* and "*Strongly Disagree"*. We are very encouraged that almost 80% answered "*Strongly agreed"* to questions on usability and ease of use. 85% answered strongly agreed to finding text relevant to their research topics in a webpage and were able to annotate the content. Only 30% agreed that they felt the need for re-annotating already annotated content in a set of webpages.55% answered agreed to question on formulating their own keywords when the existing popularity-based and vocabulary tags exhausted in the autocomplete dropdown on the first few words typed into the text box.

4. SEARCH RESULTS RETRIEVAL

Online users typically express their information needs in the form of a query, which comprises of a set of keywords submitted to a search application. The search then retrieves relevant information in the form of documents, which the system assesses to be relevant to users' information needs. Relevance here represents the similarity between the selected and suggested results.

Retrieval-oriented indexing of content in websites is at the core of our methodology and based on our earlier extensive work in [3]. We now want to look at it from the perspective of crowd-annotation and tagging and ascertain whether this layer of semantic annotation can further reduce the angle between documents and query vectors in terms of VSM. After having annotated our semantic index in the previous section, as an example, we searched for "*social research*" having semantic Named Entity (NE) containing term *"research methods*" of type "*PrintMedia*". One of the results we found in the top 10 results shows an entity "*International Social research methods case studies*" of type "*Print Media*". On closer inspection of the indexed document, we found that the full-text keywords list also lists *social research methods* as top keyword due to high score. But in the annotation element of the document index, the top annotation (free text annotation) is "*research methods bank*" and the source text (the text that has been selected for annotation in a webpage) contains "*social research methods case studies"*. So the scoring was performed based on annotated term, sourced text, entity mention and full-text keywords respectively. In comparison to the first result, when we see the second result in top 10-result set, we see that there is more annotation with the word "*research*" in it e.g. "*mobile research*", "*e-research*", "*online research links*", "*research framework*" and the DBPedia concept "*Research Methods*" but with a low score of 0.59 which was not enough for the search engine to flag this result up at no 1. That was largely due to comparatively larger similarity angles between the query terms and documents elements compared to the first result. Another interesting element in the first result is that the keywords and concepts both list "*Social research methods*" and "*social research*" as top keywords respectively in their token list, which is a cross of the original query "*social research*", and entity filter "*research methods*". This kind of heterogeneous query building (based on post-query-submission in our search application) proves to be an effective tool in retrieving most relevant search results. The *field norm* characteristics widely used in Elasticsearch in documents ranking, gives extra weight to the number of times a web document has been annotated. We will explain the score computation algorithm in Section 5.

4.1 Scalability of our Framework

The most important aspect of our annotation and searching framework is that it could be extended and used as a service as part of KB expansion. The cluster that runs the Elasticsearch node can be mounted on a dedicated server in order to serve any authenticated web server using one of the many available client

[11] http://goo.gl/MEJIze

[12] http://goo.gl/qgUQTK

[13] Link to keywords is available at http://goo.gl/QpFmpf

libraries with full community support. The front-end search application therefore doesn't necessarily have to run on a similar networked environment; rather, it can ping the Elasticsearch server as a remote server to serve online users enabling them to annotate and search using the legacy search applications.

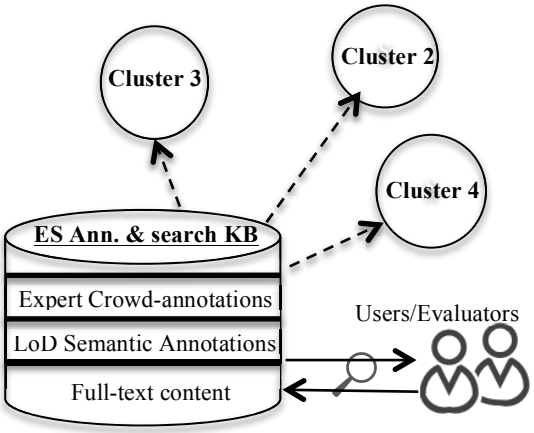

Figure 2. Extensible and scalable semantic indexing, annotation and search framework

Figure 2 actually shows one Elasticsearch cluster can be combined into multiple clusters thus making it a domain-independent, multi-disciplinary annotation and search platform available accessible to users via universal user interface.

4.2 Annotations-based Relationships

Given the following query, we can quickly discover new web documents listed in the top 10 search results based on crowd-annotations. User to user, user to web documents, experts-tagged web resources to automatically annotated web resources are a few to name when the search application extend the scoring criteria from full-text index to more meaningful elements of a document index.

```
"query": {"nested": {"path":
"crowdAnnotation", query": {"filtered":
{"query":{
"match_phrase":{"crowdAnnotation.freeTags":
"methodological innovation"}},

"filter":{"bool": {"must": [{"term":
{"crowdAnnotation.user": "user_xyz"}}]}}}}
```

Figure 3. ES query for retrieving (user-specific) annotation/tagging based results

When we look at the above query in Figure 3, we discover that we can relate various webpages based on experts' annotations, experts' research interests, webpage tagging or even the source text which is the text they select inside the webpage to superimpose their annotation on. Given that most of the participants were experts in their fields and they attempted the annotation experience very earnestly, with genuine interest in the content, we are quite encouraged with the number of related webpages annotated by multiple users.

4.3 Annotations are More Summative than Semantics and Topical Keywords Combined

When we execute the following query in Figure 4 against 3400 documents, the most relevant result in top 10 results we get is the one having been annotated and tagged with phrases "*data management*", "*data quality & management*" and "*data quality & data management*" by 3 different expert annotators. Interestingly, the list of keywords associated with the same document include "*classification variables*", "*large datasets*", "*smaller units*", "*conventions*" and concepts include "critical thinking", "want", "need" etc. Nowhere in the full text, has the document suggested "*data management*" as an activity except the title of the page where the closes phrase is "*managing your analysis*".

```
"query": {"bool": {"should": [{
"query_string": {"fields": [

"allkeywords.keywords","allentities.entity","
allconcepts.concepts"],"query": "data
management"}}, {"nested": {"path":
"crowdAnnotation","score_mode":max","query":
{"query_string": {"fields":[
"crowdAnnotation.freeTags","crowdAnnotation.a
nnotatedText"],"query": "data management"}}
```

Figure 4. ES query for *data management* keywords combining LoD-based semantic index with crowd-annotated index

In Figure 4 `keywords, entity and concepts` are LoD-generated terms and `crowdAnnotation` terms have been created by expert annotators.

What the above query in Figure 4 lacks is the connection with Typology-based (vocabulary-based) annotations of the webpages and the search results ranking will change when we modify it to the following query in Figure 5.

```
{"query": {"bool": {"should": [
{"multi_match": {"query": "data management",

"fields": ["allentities.entity",
"allkeywords.keywords",
"allconcepts.concepts"]}},

{"nested": {"path": "crowdAnnotation",

 "query": {"multi_match": {"query": "data
management","fields":
["crowdAnnotation.annotatedText",
"crowdAnnotation.freeTags"]}}}},{

 "nested": {"path": "vocabularyAnnotation",

"query": {"multi_match": {"query": data
management","fields":["vocabularyAnnotation.a
llTags","vocabularyAnnotation.narrowerTypolog
yClassification","vocabularyAnnotation.broade
rTypologyClassification"]}}}
```

(1) (2) (3)

Figure 5 ES query for *data management* keywords combining LoD-based semantic index with crowd-annotated index (including vocabulary annotation)

Now the query in Figure 5 search for terms against selected fields in non-typology and typology-based annotations (2,3) along with semantic concepts, entities and topical keywords (fields) (1). The fact that all crowd-annotation fields have less data (due to shorter and meaningful annotations) in them as compared to full-text content and title fields, they impact the retrieval scoring to a greater extent. The *field length norm* feature of the Elasticsearch scoring algorithm measures smaller field by giving them higher weighting except those modified by the *boost* factor. As we can see in the above query, the `vocabularyAnnotations.allTags`, `crowdAnnotation.annotatedText` and `crowdAnnotation.freeTags` are those fields filled up by users' annotation and tagging activity hence they carry more

weight when it comes to score calculation using the Elasticsearch standard scoring algorithms.

4.4 Semantic Information Retrieval in ES

In order to practically benefit from crowd annotations and tagging, we have developed a search application, which would enable us measure the efficacy of search results in terms of relevance and performance of the search engines at the time of indexing and retrieval. We have deployed a fully-fledged autocomplete feature on the search box in order to ascertain users' preferences at the time of query submission. The participant-based search experiments and evaluation are however beyond the remit of this paper and will be covered in the next phase of this research. The search application is currently being optimized and can be accessed at http://goo.gl/UIGGIz.

Following the submission of a query, a typical search engine, matches the query terms with indexed tokens to gather all matching documents and rank them using scoring criteria before showing the top results to user. In our case, Elasticsearch will see how relevant pages $r=\{r^1,r^2,\ldots r^n\}$ could be retrieved in top 10 pages which were retrieved against each query against full-text index $Q(k)=\{k^1,k^2,k^3..k^7\}$ and semantic index which comprises of $Q(s) = \{s^1, s2, s3\ldots s^7\}$ and $Q(c) = \{c^1, c2, c3\ldots c^7\}$ i.e. LoD-based semantic index and crowd-annotated index respectively forming one document vector in a VSM. We will talk about query-document relationship using Cosine Similarity Matrix (CSM) in the next section to highlight how best our system interpret keywords, entities, crowd-annotations at the time of search results retrieval.

4.4.1 Manipulation of Weights for Relevance Maximization

Our hybrid semantic indexing and search platform offers the flexibility of term score manipulation at query time i.e. retrieving those results having a specific named entity with the maximum score in addition to the query's terms match in other fields.

For example, the following query fetches results from the Elasticsearch KB based on users' query *team management.* In simple terms, the user wants to get all relevant documents having content on "*team management*" and the fields to search the query against include `content` (fulltext), `allconcepts, allentities, annotatedText and sourceText` (automatic & crowd terms). The filter being applied for maximum relevance is `allentities.entity` field, which must match those documents, which have entity of type "*professional development*".

```
"query": {"bool": {"should":                                    1
[{"query_string": {"fields":
["allentities.entity","content"
"allconcepts.concepts"],"query": "team
management"}},{"nested": {"path":
"crowdAnnotation", "score_mode": "max",                         2
 query": {"query_string": {"fields": [
 "crowdAnnotation.freeTags",
"crowdAnnotation.annotatedText"], "query":
"team management"}}},{"match_phrase": {
 "allentities.entity": "research team                           3
leader"}
```

Figure 6 ES query for *data management* keywords with a filter on specific Entity.

The above query retrieves results based on a cumulative score, which is 13.39 for the first result. The lowest score is 0.026, which shows variation in the maximum and minimum scores for a given query as above. The most important aspect of the above query is that the relevance score is calculated based on the 3 components, labeled 1, 2, 3 in the figure. 1 and 2 represent automatic and expert annotations respectively and 3 is a filter.

By executing the above query, we get 252 results with the top most result having 13.39 score. However, when we remove component 2 (`crowdAnnotation`) of the query and re-execute the query, we get similar results but sorted based on different score calculations. The maximum score for top result is 9.19 but we know that the score has been calculated purely based on lexical and semantic content in the index with no weight manipulation caused by expert annotations. The filter applied here is the type of entity, which could be specified by the user after the first set of results is retrieved against a given query. By including annotation component 2 in the query, the search brings up another result to the top slot with an almost similar score (13.39) but the relevance increases in terms of annotations and tagging. For example, in the above query, *team management* partially matched with `annotatedText` as well as `sourceText` but only one of the two words matched with the fulltext content of the page (no match with automatic annotations). However, since every result has to conform to the 3rd component i.e. result should have an entity of type *JobTitle* and label value "*Research team leader*", the relevance increases greatly. However, conformance of results to component 3 is not a must (due to loose filter `should`), as we prefer those results but leave it to the search engine to calculate the score based on the combination of components.

In another scenario, when we replace the keywords in the above query (Figure 6) with "*multilevel modeling"* and entity of type *Person* having label value of "*Patrick Sturgis"*, the total results produced by the search engine is 50 with maximum score of 1.75. When we look at the top result among top 10 results, we see that "*multilevel"* and "*multilevel modeling"* exist in many fields including the crowd-annotation field. However, the 3rd component doesn't conform to the name of the entity of type *Person* but the search engine has listed the page as top of the 10 pages in the results list. Removing the 2nd component from the query leads to producing 40 results with 1st results conforming to component 3 but with no presence of any crowd activity on the page whatsoever. In this case, Elasticsearch has applied the standard TF-IDF scoring algorithm to retrieve relevant search results but the page popularity in terms of crowd annotations and tags have influenced the status of the page among top 10 search results.

5. COSINE SIMILARITY BETWEEN QUERY AND DOCUMENTS

After having indexed the semantic annotation and crowd-sourced annotations in Section 3, and detailing the retrieval model in Section 4, we need to ascertain the modified ranking in terms of query, document vector similarity of *q* and *d.* Cos (*q,d*) is the Cosine of the angle between *q* and *d* to show how related are the terms in a query to a range of documents. Let's assume we have to calculate similarity of two documents in a VSM and to do that we need to convert each document to vectors, which can then be visualized in a vector space. Each document is a vector of full text terms, semantic terms and crowd annotated terms (annotation, source text and tags of free text and vocabulary). Quantification of similarity between two document

vectors and query vectors in a given vector space have to be ascertained due to the magnitude of the vector differences as two documents with very similar content may have significant vector difference simply because documentA is longer than documentB. In other words, the relative distribution of terms in two documents may be the same but the absolute term frequencies of documentA may be larger than documentB.

We have defined all English stop words not to be analyzed in the settings of our indices at the time of creation, but given that a webpage may have other terms and characters the semantic representation of which might not have been possible at the time of automatic semantic annotation. That presence may potentially result in increasing the length of documentA vector. On top of that, we assume there will still be noise emanating from users' annotation despite the fact that 90% of annotators were expert social scientists and knew from the outset the importance of annotations and tagging.

5.1 The angle of document-query relevance

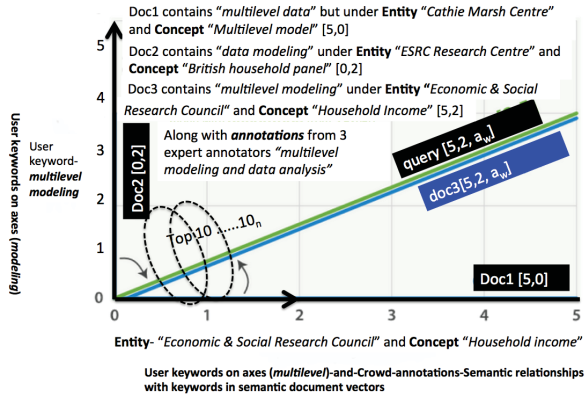

Figure 7. Two-dimensional representation of query vector in VSM

In Figure 7, we see that the user searches for "*Multilevel modeling*" but wants to filter out results based on association of content with various semantic entities and discipline-specific vocabulary. We will explain the effectiveness of TF.IDF (in our own semantic VSM model) later in this section, which Elasticsearch uses as the default way of calculating term weights for VSM and is an efficient algorithm producing high quality search results. As we can see in the figure above, a query vector representation shows weight 5 for *Multilevel* and 2 for *modeling.* Doc1 is closer in terms of smaller angle but the closest document to the query is doc3, based on other factors (a_w) in addition to mere incidental presence of words in those documents as shown in Figure above.

5.1.1 Assignment of Score to Documents

In semantic search and in the case of enhanced crowd-annotations, we need to emphasize more the context of a term than the occurrence of lexical, semantic or crowd-annotated terms in a document or collection of documents in Elasticsearch KB. Along with "how many time" the query term occurs in the document, we are interested in the "where" the term or word occurs and "how important" is it to be considered worth placing in the top 10 search results. Since we have amassed each full-text document with semantic annotations and on top of that, the expert annotators have further annotated all the content with more metadata therefore the length of documents have increased greatly. In order to measure similarity between query and document, we first need to normalize that length in order to measure the proximity of documents in now a modified VSM space. In other words, a document vector can be length-normalized by dividing each of its components by its length i.e.

$$|V| = \sqrt{\sum_{i=1}^{N(i.c.e.a)} v_i^2} \quad (3)$$

where components *i,c,e,a* represent in our case additional layers of annotation to a document vector which ES will use to calculate the score for ranked documents retrieval.

In order, to visualize a semantic document vector in a |V| dimensional vector space, we have to think of the user's query as a query vector. Document terms lie on the axes of the vector space and document vectors are points, which will be multi-dimensional in our search application. All document vectors having close proximity to query vectors in the space will be ranked higher. In terms of document vectors, we have an elongated document vector along with full text content as an Elasticsearch document. In terms of the query vector, we understand based on the data obtained from Google analytics, user queries are not abnormally lengthy but they are not single term either which will be a benefit when calculating IDF later in this section as part of Cosine similarity calculation. Most measures of vector similarity are based on the Dot product, which is given as:

$$\text{Cos}\,(q,d) = q \cdot d = \sum_{i=1}^{|V|} q_i\, d_i \quad (4)$$

5.2 Ranked Retrieval in SVS

In our web repository search, we want to retrieve the most relevant documents, which are most useful to online searchers. As we have outlined earlier, we rely on document and query vectors to measure how well documents and query match therefore, we have to look into the lengths of document vectors and get them normalized before computing cosine similarity of queries and documents vectors. For example, a document vector with Crowd-annotations will have longer lengths than those having none. However, the importance and rarity of terms will still remain important as Elasticsearch uses TF.IDF weighting distributions to compute relevance and ranks of document in top 10 search results. IDF of vocabulary tags added by the crowd-annotators and those added up by the Alchemy annotators will especially plays a role in ranking those documents higher on the scale. For example, "*British Sociological Association of Ethical Practice"* is a statement in one of the many webpages but a certain webpageA becomes more relevant when it was annotated by two expert annotators with "*BSA guide*" and a URL to the guide. Similarly another annotated the text with a free tag "*Ethical research practices*". The length of the document vector increases with the addition of these terms and phrases and tags but the rarity of the webpage has also increased for these reasons: (a) it was annotated and tagged hence IDF increases, which in turn increases overall score; (b) more contemporary data was added linking the document with more similar documents hence in the range of small cosine angles clusters in the semantic vector space; (c) words are likely to be important based on expert annotations and vocabulary-based annotations. Throughout this analysis, we consider IDF as a measure of informativeness of the term and the fact that IDF affects the ranking of documents for queries with at least two terms, which in our case is the ideal situation. For example in the above query IDF weighting makes occurrences of "*BSA*" counts far more in the document ranking than occurrences of "*guide*" for it's being common term. Also since VSM doesn't consider the

ordering of term tokens in a document so in our crowd-annotation and LoD-based semantic annotation, the order of words inside the vector stack won't matter. Rather the context, place, rarity and importance of the token will matter regardless of whether the token was generated from full text, semantic annotation or crowd-annotation inside a single document vector.

5.3 Document-Document & Document-Query Cosine Similarity Computation

Cos (q,d) or the dot product measures the cosine of the angle between q and d but the problem with dot product is that it is longer if document vector is longer in |V| dimensional vector space. The length of vector v_i (given in Equation 3) will be longer if they have higher values in each dimension which means more frequent words will have higher dot products. In our semantic vector space, we won't ideally want document query similarity to be sensitive to word frequency in the |V| dimension vector space. Elasticsearch scoring algorithm therefore normalizes the vector by dividing the two vectors by their length, hence the normalized cosine similarity between documentA and documentB is given as:

$$\cos(\overrightarrow{docA}\ \overrightarrow{docB}) = \frac{\sum_{i=1}^{|V|} docA_i\, docB_i}{\sqrt{\sum_{i=1}^{|V|} docA^2}\sqrt{\sum_{i=1}^{|V|} docB^2}} \quad (5)$$

The above equation gives us leverage to consider similarity based on factors other than only the Boolean model which results in getting relevant documents far higher in the top 10 results than the less relevant based on context (occurrence, location and importance of terms or words). The similarity angle it will give us between query and documents will lead to document ranking i.e. the smaller the angle the more relevant the document. The length-normalization of each vector by the Elasticsearch is obtained by dividing each component of a single vector by its length. That way we offset the relative distribution of terms in a set of documents and compute the proximity and relevance to the query in question. In other words, long and short documents' vectors now have comparable weights after new annotations were added automatically or by the crowd. New and contemporary tag assignment by expert annotators using the annotation tools (as explained in the previous section) thus becomes significant in terms of establishing relationships between two documents and their ranking in search results.

Similarly cosine similarity between query and document is given as:

$$Cos(\vec{q}, \overrightarrow{d}) = \frac{\vec{q}\,.\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} {q_i}^2}\sqrt{\sum_{i=1}^{|V|} {d_i}^2}} \quad (6)$$

where q_i is the TF-IDF score of term *i* in the query vector and d_i is the TF.IDF score of term *i* in the document vector. $|\vec{q}|$ and $|\vec{d}|$ here are the lengths of $\vec{q}$ and $\vec{d}$ respectively. So the normalized vector in the semantic vector space model would be equivalent to the dot product only if $\vec{q}$ and $\vec{d}$ are length normalized i.e. $Cos(\vec{q}, \vec{d}) = \sum_{i=1}^{|V|} q_i\, d_i$

5.3.1 TF.IDF Score Manipulation

$TF_{t,d}$ of term t in document d is defined as the number of times t occurs in d. We always want to compute TF when computing *query-document* match scores. A document with 10 occurrences of the term may be more relevant than a document with 1 occurrence of the term however not 10 times more relevant as relevance doesn't increase proportionally with term frequency. In order to balance the number of occurrences of repeated term in a document (in the case of repeated free-text, typology tags), we either use log frequency weight of term t in d is

$w_{t,d} = \{ 1 + \log_{10} tf_{t,d}$, if $tf_{t,d} > 0$ or simply normalize TF by dividing the number of occurrences of term t in document d by the total number of terms. Here we will use log frequency weighting (unlike ES for proof of concept) as a method of choice in order to reduce the effect of multiple occurrences of a term. DF_t on the other hand, is the document frequency of t i.e. the number of documents that contain t. in other words, DF_t is an inverse measure of the informativeness of t and $DF_t < N$. where N is number of documents in the documents collection.

The normalized TF.IDF for a document is thus computed using the following equation (7), which is the product of its normalized TF weight and its IDF weight. The IDF in our case will be based on N=3400 and DF 10 which is the optimal figure for search results in a web application. The log-weighted TF.IDF is given as:

$$W_{t,d} = 1 + \log(tf_{t,d}) * log_{10}(N/df_t) \quad (7)$$

where $idf_t = \log_{10} \frac{N}{df_t}$ and idf is the measure of informativeness of the term. To compute the sore of query-document relevance, the above equation has changed to Equation (8)

$$\text{Score (q,d)} = \sum_{t \in q \cap d\{k,c,e,\forall Ann\}}^{N} 1 + \log tf_{t,d} \,.\, \log \frac{N}{df_t} \quad (8)$$

The score is 0 if none of the query terms is present in the document. ES will obviously give more weight to rare terms, as they are more important than frequent terms hence increased IDF in the above equation. The IDF component increases with the rarity of the term in the *document collection* but it also increases with the number of *occurrences* within a document thereby increasing the length of that document vector.

5.4 Search Results Retrieval using CS

Table 1 shows a vector representation of three documents in terms of real-valued vector of TF-IDF weights $\in \mathbb{R}^{|V|}$ calculated using (5).

Table 1. Log frequency TF.IDF weights of non-Semantic Document Vectors {UN →N}≡ {*Un-normalized to Normalized*} using Equation (5) for document vectors.

Query terms	Doc1 {UN →N}	Doc2{UN →N}	Doc3 {UN →N}
methodology	0.52	0.38	0.64
longitudinal	0.59	0.55	0.49
ethical	0.38	0.46	0.30
social	0.41	0.24	0
childhood	0	0.35	0.37
framework	0	0.21	0
socio-economic	0.20	0.10	0
stakeholders	0	0.16	0.20
dissemination	0.10	0.24	0.26

The $tf_{t,d}$ of documents were initially represented in count vector matrix (using 7) but to sum up the calculations we will only use log frequency weighted TF-IDF of various terms in three different documents. We will make comparison of weighted document vectors i.e. non-semantic document vectors and

semantic document vectors. We will then compute cosine similarity between the query vector and each semantic and non-semantic document vectors.

In order to compute cosine similarity between query and document vectors, we need to represent query q in terms of a vector. A query "*Socio-economic policy framework*" will thus be converted into a count vector model (unlike weighted document vector) and using Equation (6), cosine similarity will be calculated as below:

Table 2: Cosine similarity computation between query documents in a non-semantic vector space model using (6)

Query {*N=3400*}				Doc1		Doc2		Doc3	
Query terms	W. $tf_{t,d}$	DF	IDF	NW	DP	NW	DP	NW	DP
Socio-economic	1	2	3.23	0.20	0.64	0.10	0.32	0	0
policy	1	0	0	0	0	0	0	0	0
framework	1	1	3.53	0	0	0.21	0.74	0	0
Final Similarity score between *q* & *d*				0.64		0.32+0.74=1.06		0	

NW=Normalized weight, DP=Dot product, W.=Weighted

As we can see from Table 2, the similarity score between doc2 and query is 1.06, which suggests that, the document is closely related to the query terms after normalizing the length of the document.

To compute the similarity score based on semantic document vectors, we have annotated all the three documents with relevant phrases in which some terms include *policy*, *framework*, but no *socio-economic* term. Here is the modified Table 1 labeled as Table 1a.

Table 1a: Log frequency TF.IDF weights of Semantic Document Vectors {UN →N}≡ {Un-normalized to Normalized} using Equation (5) for document vectors

Query terms	Doc1 {UN →N}	Doc2 {UN →N}	Doc3 {UN →N}
methodology	0.51	0.37	0.62
longitudinal	0.58	0.55	0.47
ethical	0.38	0.46	0.29
social	0.41	0.24	0
childhood	0	0.35	0.36
framework	0.10	0.21	0.20
socio-economic	0.20	0.11	0
stakeholders	0	0.17	0.20
dissemination	0.10	0.24	0.25
policy	0	0.11	0.13

As we can see that *policy* term has been added to Doc2 and Doc3, *framework* to Doc 1 and Doc 3, which have changed the normalized TF-IDF weighted score of semantic document vectors. Now after computing the cosine similarity between query and document, vectors we get different scores as shown in Table 2a.

We can obviously see in Table 2a that Doc2 has gone further higher in ranking score but Doc3 is now runner up in the list and has pushed down Doc1 in the ranking, which was quite expected given that *framework* was part of the annotated terms in both Doc1 and Doc2.

Table 2a: Showing revised weights after new terms were added through semantic or crowd-annotation (using (6))

Query {N=3400}				Doc1		Doc2		Doc3	
Query terms	W. $tf_{t,d}$	DF	IDF	NW	DP	NW	DP	NW	DP
Socio-economic	1	2	3.23	0.20	0.64	0.11	0.35	0	0
policy	1	0	3.23	0	0	0.11	0.35	0.20	0.64
framework	1	1	3.05	0.10	0.30	0.21	0.64	0.13	0.39
Final Similarity score between *q* & *d*				0.64 +0.305= 0.95		0.35+0.35 +0.64= 1.34		0.64+0.39= 1.03	

Such phenomenon impacts the overall score more sharply if the annotation was a vocabulary term instead of a free text word, which may contain more *stopwords* or repeated words. The IDF in the case of vocabulary terms/tags in annotations will increase on the basis of rarity of terms in the collection of documents thus making the document or set of documents more relevant for top 10 search results in a web repository search.

6. CONCLUSION & FUTURE WORK

We recognize that based on our insight into the web of semantic indexing, crowd-annotation and searching, that this area of research continues to evolve with the fast-paced information revolution. Automation of 'semanticizing' scientific data inside today's web for the sake of their consumption in the web search of tomorrow may not be possible in entirety but it does offer promising results when used in conjunction with the involvement of the consumers of that data. In other words, there is greater need for community of online users especially researchers to aid the search engines in determining the degree of relevance of a desired piece of information at the time of searching. Willingness to contribute in terms of annotating and tagging content in a multi-disciplinary research data repository alongside the consumption of information will go a long way in terms of relevant and precise information retrieval. In order to continue with this work, we intend to further expand the annotation and tagging environment by including a number of web repositories predominantly containing scientific content having a high and active online user base. We will also continue to work on the search application to evaluate the search results based on our retrieval and cosine similarity model explained in Section 4 and 5. The next phase of this research will enable us to delve more deeply into interpreting annotations and tags of the research community and ascertain its impact on the relevant search results retrieval.

7. REFERENCES

[1] Fernandez, M., et al., *Semantically enhanced Information Retrieval: An ontology-based approach.* Web Semantics: Science, Services and Agents on the World Wide Web, 2011. **9**(4): p. 434-452.

[2] Wu, P., A. Heok, and I. Tamsir, *Annotating the Web Archives – An Exploration of Web Archives Cataloging and Semantic Web*

Digital Libraries: Achievements, Challenges and Opportunities, S. Sugimoto, et al., Editors. 2006, Springer Berlin / Heidelberg. p. 12-21.

[3] Khan, A., T. Tiropanis, and D. Martin. *Exploiting Semantic Annotation of Content with Linked Open Data (LoD) to Improve Searching Performance in Web Repositories of Multi-disciplinary Research Data*. in *9th Russian Summer School, RuSSIR 2015, Saint Petersburg, Russia, August 24-28, 2015*. 2015. Springer International Publishing.

[4] Mirizzia, R., A.R.T. Di Noiaa, and E. Di Sciascioa, *Lookup, Explore, Discover: how DBpedia can improve your Web search.* 2010.

[5] Riggs, F.W., *Interconcept report: a new paradigm for solving the terminology problems of the social sciences*. Vol. 44. 1981: Unesco.

[6] Snow, R., et al. *Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks*. in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2008. Association for Computational Linguistics.

[7] Royo, J.A., et al. *Searching the Web: from keywords to semantic queries*. in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*. 2005.

[8] Shabanzadeh, M., M.A. Nematbakhsh, and N. Nematbakhsh. *A Semantic based query expansion to search*. in *Intelligent Control and Information Processing (ICICIP), 2010 International Conference on*. 2010.

[9] Wu, X., L. Zhang, and Y. Yu. *Exploring social annotations for the semantic web*. in *Proceedings of the 15th international conference on World Wide Web*. 2006. ACM.

[10] Zervanou, K., et al., *Enrichment and Structuring of Archival Description Metadata.* ACL HLT 2011, 2011: p. 44.

[11] Yang, C., K.-C. Yang, and H.-C. Yuan, *Improving the search process through ontology-based adaptive semantic search.* The Electronic Library, 2007. **25**(2): p. 234-248.

[12] Bao, S., et al., *Optimizing web search using social annotations*, in *Proceedings of the 16th international conference on World Wide Web*. 2007, ACM: Banff, Alberta, Canada. p. 501-510.

[13] De Virgilio, R., *RDFa Based Annotation of Web Pages through Keyphrases Extraction*

On the Move to Meaningful Internet Systems: OTM 2011, R. Meersman, et al., Editors. 2011, Springer Berlin / Heidelberg. p. 644-661.

[14] Khan, A., D. Martin, and T. Tiropanis, *Using Semantic Indexing to Improve Searching Performance in Web Archives*, in *International Journal on Advances in Internet Technology*. 2012: Seville, Spain. p. 1-4.

[15] Bontcheva, K., V. Tablan, and H. Cunningham, *Semantic Search over Documents and Ontologies*, in *Bridging Between Information Retrieval and Databases*, N. Ferro, Editor. 2014, Springer Berlin Heidelberg. p. 31-53.

[16] Benjamins, R., et al., *The six challenges of the Semantic Web.* 2002.

[17] Halpin, H., *Social Semantics: The Search for Meaning on the Web*. Vol. 13. 2013, USA: Springer US. 220.

[18] Cappiello, C., et al. *A Quality Model for Linked Data Exploration*. in *International Conference on Web Engineering*. 2016. Springer.

[19] Gangemi, A., *A Comparison of Knowledge Extraction Tools for the Semantic Web*, in *The Semantic Web: Semantics and Big Data*, P. Cimiano, et al., Editors. 2013, Springer Berlin Heidelberg. p. 351-366.

[20] Rizzo, G., et al., *NERD meets NIF: Lifting NLP Extraction Results to the Linked Data Cloud.* LDOW, 2012. **937**.

[21] Gabrilovich, E. and S. Markovitch. *Computing semantic relatedness using Wikipedia-based explicit semantic analysis*. in *IJcAI*. 2007.

[22] Burghardt, M. *Usability recommendations for annotation tools*. in *Proceedings of the Sixth Linguistic Annotation Workshop*. 2012. Association for Computational Linguistics.