Random indexing spaces for bridging the Human and Data Webs

Jose Quesada, Ralph Brandao-Vidal, Lael schooler

Max Planck Institute, Adaptive Behavior and Cognition, Berlin
Lentzeallee 94, 14195 Berlin
{quesada, rbrandao, schooler}@mpi-berlin.mpg.de

Abstract. There exists a wide gap between the information that people and computers respectively can operate with online. Because most of the web is in plain text and the Semantic Web requires structured information (RDF), bridging the two worlds is an important current research topic. Here we propose a web service that uses a Random Indexing (RI) semantic space trained on the plain text of the one million most central Wikipedia concepts. The space provides us with vectors for each of the equivalent DBpedia concepts and vectors for any text or webpage. It can also provide a hashed version of the RI vector that works as unique handler like URIs do, but with the additional advantage that it represents text meaning. As a result, any page (previously readable only for humans) is now integrated with the Semantic Web graph using links to one of its most central parts, DBpedia.

Keywords: text mining, statistical semantics, structured information, identifiers, resources, literals, RDF

1 Introduction

Most of the existing knowledge on the Web is in plain, unstructured text[1]. That is, most web pages contain data expressed in a way that is easily understandable for humans but hard to interpret for machines. The Semantic Web promises large interoperability gains, but it all depends on how well we can integrate two separate worlds. On the one hand we have rich structured datasets following linked data principles [1] with the ultimate goal of being able to use the Web like a single global

[1] Governments, enterprises and almost any dynamic website all have large bodies of knowledge already in structured form (relational databases) but not following linked data principles. Converting it into RDF is an interesting problem, but not directly related to the problem we discuss here (how to find the closest DBpedia concepts to any text passage) so we will not elaborate further.

database. But while the Semantic Web graph is growing at a very healthy rate[2], it is still a marginal part of the entire Web. On the other hand we have the flat, messy (but abundant) plain text Web pages. Traditional information retrieval and machine learning techniques that work on plain text have been making steady progress for some time now. Some of these techniques use structured data too [2].

The problem we aim to solve in this paper is simply converting literals into resources. This problem is trivial if the only requirement is a unique ID (a random one would suffice). But giving unique IDs is not a solution; to integrate the new resource we need to generate outwards links to the rest of the Semantic Web graph. That is, we enhance the meaning of the new node by generating new connections. We achieve this thanks to statistical semantics on a corpus that has parallel representations of both worlds: Wikipedia/DBpedia [3]. Random indexing (RI) [4] offers a highly scalable way of assigning semantic vectors to Wikipedia concepts. We then compress the vectors using MD5 hashing and use these hashes as meaningful identifiers that become part of the RDF graph.

For clarity, we will refer to the Semantic Web and linked data initiative as the data Web. The human Web is simply the current Web made of pages and unnamed links.

1.1 An overview of URIs, Resources and literals

We build on three basic notions: URI, resources, and literals. In summary, URIs are unique identifiers, and resources differ from literals in that they have URIs and can link to other nodes in the graph. We will describe these three concepts next.

1.1.1 Uniform Resource Identifier (URI)

A Uniform Resource Identifier (URI), according to the specification [5], is a compact sequence of characters that identifies an abstract or physical resource. Valid URIs take the following form:

Scheme ":" ["//" *authority* "/"] [*path*] ["?" *query*] ["#" *fragment*]

Uniform Resource Locators (URLs) are a subclass of URI, subject to the same grammar. The main difference is that a URL must point to specific information, usually a file that can be displayed on a browser or downloaded, whereas URIs do not need to[3]. People who have been on the internet for years now are completely used to this grammar. Note that none of the parts are particularly informative to describe the resource they point to. *Authority* is perhaps informative because it could carry the name of the entity (company, person, association, etc) that hosts the page. In recent years RESTful services [6] make *Paths* describe the actions they perform (e.g. read, delete, etc). The title of the page can also be part of *Path*, and some popular software such as *WordPress* implements this policy by default. However, these are all usage

[2] New datasets are added constantly to the W3C site 'Linking Open Data' http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets and the existing ones keep growing.

[3] But it is a good practice to make URIs point at some description of what they are.

conventions, but not enforced by the URI design. There is nothing in the scheme that says URIs should be meaningful for humans or machines[4].

The main role of a URI (and only requirement) is to provide a unique identifier for a resource. In this paper we will propose that it is desirable to make identifiers meaningful for machines, in a way that uses human similarity judgments.

1.1.2 Resource

The first explicit definition of resource is found in RFC 2396 [7] and states that *A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources.*

The concept of resource is primitive in the Web architecture, and is used in the definition of its fundamental elements. The term was first introduced to refer to targets of URLs, but its definition has been further extended to include the referent of any Uniform Resource Identifier in RFC 3986 [5]. That is, the concept started in the human Web, and grew to be used in the data Web. A resource is simply anything that can be identified with a URI. Note that the concept of URI contains the URL as a special case.

Resources can have properties. For example, the resource 'FidoTheDog' may have the Name property 'Fido'. That is, resources can link to other resources and to literals.

1.1.3 Literals

Literals are values that do not have a unique identifier. They are usually a string that contains some human-readable text, for example names, dates and other types of values about a subject. In the previous example, the string 'Fido' is a literal. They optionally have a language (e.g., English, Japanese) or a type (e.g., integer, Boolean, string), but this is about all that can be said about literals. They cannot have properties like resources. Unlike resources, literals cannot link to the rest of the graph. They are second-class citizens on the Semantic Web. In terms of graphs, literals are one-way streets: since they cannot be the subject of a triple, there can be no outgoing links to other nodes.

[4] In practice, URLs do have some meaning for humans, but mostly due to cues acquired after years of using them. Short, meaningful names are better, and of course more expensive, so they hint that the owner must have made a serious investment and thus be committed to the content.

1.2 Why turning literals into resources is useful

Consider that human Web nodes (pages) are literals once they merge with the data Web. The number of literals in the joint graph will be enormous, considering that the human Web is several orders of magnitude larger than the current data Web. But the number of new nodes is not necessarily just the number of webpages: a selection of text, say a paragraph, can also become a literal.

We offer a method to transform literals into resources. This solution we propose here is one of many possible: for example, there are efficient tools, such as openCalais, that link entities to semantic Web concepts using named-entity recognition. The key difference is that named-entity recognition links individual words to existing resources, where as we create a URI for larger chunks of text, such as a sentence, a paragraph or entire webpage. Like openCalais, we link the resulting URI's to DBpedia [3], one of the most central datasets in the Semantic Web.

In the next subsections, we show advantages to turning literals into resources from a graph machine learning point of view.

1.2.1 Increased integration of the human and data Webs

The current state is that even though the two Webs are essentially separated, there is some integration in at least two fronts. First, semantic Web URIs resolve into something a human with a browser can see (e.g., plain text description of an object). This is a good practice, but not enforced. Second, recently more and more parts of the human Web carry snippets of structured information (RDFa). Only recently have webmasters started using RDFa. Search engines such as Yahoo and Google are indexing RDFa too.

Integration is challenging because the two webs are structurally very different. The semantic Web is a directed labeled graph, whereas the 'human' Web is a directed unlabeled graph[5]. To merge them, we would need to produce labels for unlabeled links. But this is a problem because links in the human Web, by design, do not have labels. We could use a homogeneous label name (something like 'links-to') but then 'links-to' would become the most frequent label, eclipsing every other one and making the resulting graph harder to do reasoning with. An example of this generalist label is the 'wikilink' predicate in DBpedia. Wikilinks are the simplest links from one Wikipedia article to another. They are parsed from Wikipedia articles bodies for DBpedia as simple "source page" and "destination page" pairs. Compared to the other kinds of RDF triples in DBpedia, they are the most general, in the sense that they cover the most kind of relations, yet are the least precise, because they don't have a relation property, only using a generic "wikilink" relation type. There are 70 millions Wikilink triples, compared to 30 million Infobox dataset triples or only 7 millions Wikipedia Categories dataset triples. In such a large proportion, unnamed links would

[5] Directed labeled graphs are a lot harder to work with than unlabeled graphs, and the algorithms that work on directed labeled graphs are but a portion of all graph algorithms.

overpower the named ones for some tasks (e. g. [8]) and their addition would be detrimental.

1.2.2 Dangling nodes

One important feature of RDF is that a literal may be the object of an RDF statement, but not the subject or the predicate. Because a resource offers richer possibilities to Semantic Web practitioners compared to a literal, the joint graph would be better served having as many resource nodes as possible. From the point of view of graph theory, literals are 'one-way street' nodes that can be problematic. A node that receives connections but never links outwards is called a 'dangling node'. So literals are dangling nodes. Operating on a graph with a high proportion of dangling nodes makes some useful algorithms slower (e.g., finding shortest paths), and some other harder to use or impractical. For example, straight pagerank has problems with dangling nodes, even though in practice they can be solved [9], but other algorithms such as singular value decomposition require a matrix with no all-zero rows (a dangling node produces an all-zero row).

One alternative is to remove dangling nodes. Some studies that look for shortest paths remove literals because dangling nodes would add one-way-streets and search would take longer [10]. But this has unintended secondary effects. Removing the dangling nodes somewhat skews the results on the non-dangling nodes since the outdegrees from the non-dangling nodes are adjusted to reflect the lack of links to dangling nodes.

The Semantic Web graph has a large proportion of dangling nodes. According to data reported in the landing page of the Linked-Data Semantic Repository (LDSR, [11] including DBpedia, Freebase, Geonames, UMBEL, Wordnet, CIA World Factbook, Lingvoj, MusicBrainz and others), 39% of the nodes are literals (see table 1). This is the proportion of literals over the total number of entities. LDSR is not the Semantic Web's entire graph, but we would expect to find a similar distribution of URIs vs. literals if we could access equivalent statistics for every subcomponent.

Table 1. Statistics from the linked-data semantic repository (LDSR, [11], retrieved 3-4-2010)

Number of URI:	126,875,974
Number of Literals:	227,758,535
Total number of entities:	354,635,159

Reducing the proportion of literals compared to resources on the Semantic Web graph may open the door to better machine learning algorithms. We will explore this idea in the next section.

2 How to create Identifiers that are not only unique, but meaningful

Here we use statistical semantics to create meaningful identifiers for literals. We term these meaningful unique identifier (MUID, pronounced "mood"). We propose that algorithm for generating MUID's should have the following properties:

1. A MUID should have some (primitive) form of compositionality. If we generate a MUID for part of a page, the part's MUID should be similar to that of the full page.

2. If two pages get similar MUIDs, they should be perceived as similar by human observers.

3. Changes that are perceived as incremental by people (e.g., a blog post getting comments), should result in incremental changes to the corresponding MUID's corresponding to before and after the changes.

To understand how our proposal implements these requirements, we next describe statistical semantics, focusing on Random Indexing (RI) [15].

2.1 Statistical semantics

Statistical semantics is a general category of machine learning algorithms that exploits statistical patterns of human word usage to figure out word meaning. These algorithms come from cognitive science and information retrieval. A typical task for statistical semantics is to measure the semantic similarity of two passages. The answer is given as a number, usually the cosine between the vectors that represent the passages in some high dimensional space. The vector for a passage is usually the average of the vectors for all the words in it. The vectors for each passage, when averaged together, form the document vector. This implements compositionality (property 1 above) and addresses incremental changes (property 3), because recomputing a vector when the text is only slightly different will produce only a slightly different vector.

Most statistical semantics methods start with a frequency matrix of word by documents [12], and many apply different transformations to these matrix (example, truncated singular value decomposition). The vector space model [12] was the first of these methods. It improves over Boolean information retrieval (IR) in that it allows computing a continuous degree of similarity between queries and documents, and this makes ranking possible. It also moved IR from set theory to linear algebra, which facilitated the explosion of newer models. These newer models such as LSA and random indexing extend the approach by adding generalization, that is, these models are able to tell when two words are synonyms. Table 2 shows an example. For an exact matching algorithm based on a Boolean vector space, the similarity between pairs of words is all-or-nothing. In contrast, newer models such as LSA and RI capture the similarity of doctor to physician and surgeon.

Table 2. Generalization. How LSA solves synonymy. Cosine values from lsa.colorado.edu

	Boolean/vector space	LSA
doctor – doctor	1	1
doctor – physician	0	.8
doctor – surgeon	0	.7

Statistical semantic models are trained on a large corpus of text that is representative of the domain of interest. Once this space has been created, it can be used to compare not only passages from the training corpus, but novel passages as well. For general knowledge, the corpus used to be an encyclopedia or a sample of textbooks representative of what a college student would have read [13]. A corpus composed of traditional encyclopedias or textbooks have significant limitations. First many recently-coined terms common on the web, such as iPad, are not in those datasets. And second, there's no direct mapping between these corpora and resources in the semantic web. These limitations lead us to use Wikipedia as a training corpus. In the combination of Wikipedia and DBpedia we have exactly what we need, a parallel corpus that exists in both the human and data Webs.

Starting with the full text of a recent (March 2008) Wikipedia dump, we selected the most central concepts by dropping those with fewer than five in links or fewer than five out links. We applied other basic preprocessing steps described in [14]. The initial parsing produced close to a million types; as expected from natural, unedited text, most of these were typos. We then dropped types that occurred less than 10 times, and those that appeared in less than 10 documents. Approximately half of the types went away. After parsing the Wikipedia XML dump, we obtained 2.7 Gb of text in 1,279,989 articles.

3 Random indexing

Our application of Random Indexing [4] starts with the same words by documents matrix described above, taking a document to be a Wikipedia article. Then each word and each context is first assigned a random high-dimensional sparse vector: they are seeded with a small proportion of ones and negative ones with all other elements set to zero.

Once the sparse binary index vectors are constructed, a word's vector becomes the sum of the vectors for the contexts in which it appears throughout the text corpus. Conversely, a document space can also be constructed as the sum of the index vectors for words appearing in each document. Random Indexing depends on the term-document matrix computed from a corpus being sufficiently sparse that vector representations can be projected onto a basis comprising a smaller number of randomly allocated vectors. Due to the sparseness condition, the basis of random vectors has, in general, a high probability of being orthonormal. That is, every random vector will be orthogonal to any other random vector. The most exhaustive

description of RI is [15]. The main advantage of RI compared to LSA [16] is its scalability. The SVD is a computationally expensive operation. It needs to place large matrices in memory, and it may take days to compute for a dataset the size of LDSR, if at all possible. RI does not have large memory requirements and the linear algebra operations are simpler and faster.

We used the semantic vectors library [17]. This library has been proved to scale well: Cohen et al. [18] use it in an experiment with 15M documents from MedLine.

We manipulated the following parameters (see first two rows in table 3): (1) Number of dimensions. This is simply the size of the random vector that represents a word. It has the largest influence on how long it takes to compile a space, and how much storage it needs. Surprisingly, there is little published on how to select the optimal dimensionality. We manipulated dimensions from 800 to 1200. (2) Nonzero seed values. This parameter is not commonly reported in the literature. However, we found that it does change results, so we manipulate it here systematically.

Our assumption is that the parameters that work best on traditional psychological tasks will also work well for our current task of getting the most meaningful neighbors on a Wikipedia space. In the next section we try to obtain the best parameters for our web service using four well-known human similarity datasets.

3.1 Results

We used the following datasets for word pair similarity judgments: Rubenstein and Goodenough (1965) [19], Miller and Charles (1991) [20], Resnik [21] (1995; this is a replication of Miller and Charles) and Finkelstein et al. (2002) [22]. An example of the materials on these tasks would be 'How similar are gem and jewel?' Participants produce ratings going from zero (not related at all) to four (perfect synonymy).

Table 3 shows how models, based on different parameterizations of RI, correlate with the human judgments in these four datasets for word-word comparisons.

Since average human agreement in tasks like these is around .6, our results are acceptable, even though they are below some other published results [23]. For the web service, we kept the space with 1000 dimensions and 700 seed values, which seems to do well across datasets.

4 The Web service

The interface to the Web service is described using WSDL [24]. The Web service takes either a URL or plain text. When taking a URL, it parses the page and extracts the plain text. The text is then transformed into a RI vector by retrieving vectors for all its terms and averaging these together. What we provide here is a prototype that

takes about 2 min to process a request. Fortunately, the algorithm is parallelizable. An interface for testing can be reached at: http://mpi-ldsr.ontotext.com/webservice6.

Table 3. RI correlation to the human gold standard in four datasets for word-word comparisons. The average human agreement is around .6. Best results are bold.

dims	seed	Miller	Resnik	Rubenstein	wordsim
800	500	0.39	0.46	0.42	0.35
	600	0.61	0.54	0.52	0.4
	700	0.57	0.56	0.48	0.35
	750	0.37	0.44	0.39	0.34
1000	300	0.5	0.46	0.42	**0.4**
	700	**0.55**	**0.6**	0.5	0.39
	800	0.42	0.47	0.46	0.36
	900	0.48	0.5	0.42	0.37
	950	0.6	0.55	**0.53**	0.37
	980	0.49	0.43	0.4	0.36
1200	500	0.53	0.55	0.53	0.38
	900	0.5	0.56	0.47	0.36
	1000	0.47	0.51	0.46	0.36
	1050	0.34	0.5	0.41	0.37
	1100	0.42	0.45	0.38	0.37
1800	1500	0.43	0.43	0.5	0.37
2000	1900		0.57		

The web service returns both a list of nearest neighbors in the space and a unique, meaningful ID (MUID). Table 4 shows an example of the concepts that the Web service produces for an interview with Shane Simonsen[8], a UK computer science professor who abandoned the University system. The ten closest neighbors are related to education in different parts of the world, which reflects the gist of the text.

What follows is the N3-formatted RDF that this text would return for the first item in the example. The kind of links we generate are essentially unlabeled (as discussed in the introduction), but right now we use `skos:related` to express the fact that the input text is related to the DBpedia concept listed.

```
  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
  @prefix mpib <http://mpi-ldsr.ontotext.com/mpib#'> .
  @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
  @prefix DBpedia: <http://en.wikipedia.org/wiki#> .
  mpib:39f2ea57cf982d7eedccf28f92ebf13f             skos:related
dbpebia:Education_in_the_People's_Republic_of_China> .
```

[6] Alternatively http://93.123.21.85:8087/ri-webservice/
[8] http://www.lambdassociates.org/blog/interview.htm

Since 1000-dimensional vectors are too long to be used as metadata, we return a hashed version of the vector compacted with the hashing function MD5 digest.

Table 4. A sample paragraph from submitted text (left) and top 10 DBpedia concepts that the web service produced.

Your article "Why I am not a Professor", outlining your departure from academia in 1999 over declining standards and conditions, was written in 2007. Can you shed light on any further changes of the state of the tertiary education system since then? There is a recognition at government level that the standards have dropped at university and that degree inflation is rife. The UK government has abandoned its target of 50% of the population in higher education. The public sector deficit has caused the university budget to be cut by £500 million in 2010 and we shall see further cuts. However all the mechanisms of assessment discussed in that essay are still in place.	Education in the People's Republic of China Education in the United States Community college Tertiary education in Australia Education in South Korea Business-education partnerships Unemployment Secondary education in Japan Education in Thailand Education in England

5 Discussion and conclusions

We have presented a method, reachable as a web service, to attach meaning to a resource that locates it in a semantic space. Using statistical semantics we integrate any plain text literal (a paragraph or an entire page) with DBpedia, one of the central components of the Semantic Web. When literals are passed to our web service they receive a Random Indexing vector and a list of links to the 10 closest DBpedia concepts. This Random indexing vector is taken as a meaningful, unique ID (MUID) that can be used to refer to this newly-created resource. These MUIDs serve not only as unique identifiers, but as well add functionality. In the same way that in the physical world coordinates enabled location-aware applications, the semantic annotation of literals enables new functionality, such as defining the similarity of pairs of objects, and finding the most similar resources. But there is a critical difference between semantic and physical spaces. Whereas the physical world has 3 dimensions, the semantic world, as we have proposed here, may have thousands.

6 References

[1] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data–the story so far,” *International Journal on Semantic Web and Information Systems*, vol. 5, 2009, pp. 1-22.

[2] Renaud Delbru, Nickolai Toupikov, Michele Catasta, Robert Fuller, and Giovanni Tummarello, “SIREn: Efficient search on semi-structured

documents," *Lucene in Action*, Manning Publications, 2004.
[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the web of data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, 2009, pp. 154-165.
[4] M. Sahlgren, "An introduction to random indexing," *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, Citeseer, 2005.
[5] T. Berners-Lee, R. Fielding, and L. Masinter, "RFC 3986: Uniform resource identifier (uri): Generic syntax," *The Internet Society*, 2005.
[6] L. Richardson and S. Ruby, *Restful Web Services*, O'Reilly Media, 2007.
[7] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifiers (URI): generic syntax," 1998.
[8] O. Liu, "Relation Discovery on the DBpedia Semantic Web," 2009.
[9] I.C.F. Ipsen and T.M. Selee, "PageRank computation, with special attention to dangling nodes," *SIAM J. Matrix Anal. Appl*, vol. 29, 2007, pp. 1281–1296.
[10] J. Lehmann, J. Schüppel, and S. Auer, "Discovering unknown connections–the DBpedia relationship finder," *1st SABRE Conference on Social Semantic Web (CSSW)*, 2007.
[11] A. Kiryakov, D. Ognyanoff, R. Velkov, Z. Tashev, and I. Peikov, "LDSR: a Reason-able View to the Web of Linked Data," *International Semantic Web Conference (ISWC)*, 2009.
[12] G. Salton, A. Wong, and C.S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, 1975, p. 620.
[13] S.M. Zeno, S.H. Ivens, R.T. Millard, and R. Duvvuri, *The educator's word frequency guide*, Brewster, NY: Touchstone Applied Science Associates, 1995.
[14] E. Gabrilovich and S. Markovitch, "Wikipedia-based semantic interpretation for natural language processing," *Journal of Artificial Intelligence Research*, vol. 34, 2009, pp. 443-498.
[15] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, 2009, pp. 139-159.
[16] T.K. Landauer and S.T. Dumais, "A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, 1997, pp. 211-240.
[17] D. Widdows and K. Ferraro, "Semantic vectors: a scalable open source package and online technology management application," *Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.
[18] T. Cohen, R. Schvaneveldt, and D. Widdows, "Reflective Random Indexing and indirect inference: A scalable method for discovery of implicit connections," *Journal of Biomedical Informatics*, 2009.
[19] H. Rubenstein and J.B. Goodenough, "Contextual correlates of synonymy," *Communications of the As- sociation for Computing Machinery*, vol. 8, 1965, pp. 627-633.
[20] G.A. Miller and W.G. Charles, "Contextual Correlates of Semantic Similarity.," *Language and cognitive processes*, vol. 6, 1991, pp. 1-28.
[21] P. Resnik, "Using information content to evaluate semantic similarity in a

taxonomy," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 1, 1995, pp. 448-453.

[22] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, and G. Wolfman, "Placing search in context: The concept revisited," *ACM Transactions on Information Systems*, vol. 20, 2002, pp. 116–131.

[23] M.N. Jones and G. Recchia, "Scalable Techniques for Creating Semantic Vector Representations," *under review*, 2010.

[24] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, *Web services description language (WSDL) 1.1*, 2001.