Getting the Meaning Right: A Complementary Distributional Layer for the Web Semantics*

Vít Nováček, Siegfried Handschuh, Stefan Decker

Digital Enterprise Research Institute (DERI)
National University of Ireland Galway (NUIG)
IDA Business Park, Lower Dangan, Galway, Ireland
E-mail: vit.novacek@deri.org

Abstract. We aim at providing a complementary layer for the web semantics, catering for bottom-up phenomena that are empirically *observable* on the Semantic Web rather than being merely *asserted* by it. We focus on meaning that is not associated with particular semantic descriptions, but emerges from the multitude of explicit and implicit links on the web of data. We claim that the current approaches are mostly top-down and thus lack a proper mechanisms for capturing the emergent aspects of the web meaning. To fill this gap, we have proposed a framework based on distributional semantics (a successful bottom-up approach to meaning representation in computational linguistics) that is, however, still compatible with the top-down Semantic Web principles due to inherent support of rules. We evaluated our solution in a knowledge consolidation experiment, which confirmed the promising potential of our approach.

1 Introduction

The Semantic Web has been designed for asserting meaning of things mostly in a top-down manner (via explicit specifications of RDF descriptions or ontologies). We conjecture that there is also another, bottom-up meaning of the web (both the 'semantic' and 'human' one). Similarly to the meaning of natural languages arising from the complex system of interactions between their individual speakers [1], we conceive the bottom-up web semantics as consisting of implicit patterns. In the Semantic Web case, though, the complex patterns of meaning emerge from a simple language of countless triple statements, which may come from the evolving Linked Open Data cloud, but also from the human web (mediated to machines by methods like data or text mining).

The proposed alternative way of looking at the Semantic Web can bring better solutions to problems in areas like knowledge consolidation (by which we basically mean clustering of related entities and properties). For instance, in our

* This work has been supported by the 'Líon II' project funded by SFI under Grant No. SFI/08/CE/I1380. We are indebted to Ed Hovy (ISI, USC), who had an indirect, yet substantial influence on the presented research. Also, we would like to thank to Václav Belák (DERI) for discussions about possible interpretations of our results.

CORAAL prototype (see `http://coraal.deri.ie`), users can search for properties linking particular life science entities (like genes or diseases). CORAAL extracts all the underlying statements automatically from text, which leads to thousands of properties occurring only in very small number of triples. This may result in too specific query answers and user frustration, as they have to struggle to figure out how to get more general information. Imagine one wants to know more about organs involved in the production of a hormone H. A query for that can look like *H secreted_in ?x*. However, such a query may retrieve only a single result. More results could be retrieved via related properties like *excreted_in* or *produced_in*, but it is rather tedious to try all such possibilities without knowing precisely how exactly one should ask. A solution grouping extracted content into more general inter-related clusters would significantly improve the user satisfaction and efficiency, as hitting a single property would also reveal all the related ones. Yet for achieving such a consolidation, one needs to know not (only) what is meant by the statements at the level of the particular documents (which is covered by the current approaches). What is more important (and less explored) are the minuscule contextual features distributed across the whole data set (e.g., properties and $\langle$ subject, object $\rangle$ tuples that tend to co-occur at a larger scale with sufficient significance). This is what constitutes the global evidence of what is actually *meant* by the data set at large (and not just *asserted* at the level of local semantic descriptions). By capturing these aspects, one can consolidate the little scattered chunks of related knowledge in an empirically valid manner. As detailed in Section 2, we lack a comprehensive solution for this, though.

Therefore we have proposed (in Section 3) a framework stemming from recent advances in distributional semantics. This sub-field of computational linguistics is based on a hypothesis that "a word is characterized by the company it keeps" [2]. In our case, we can rephrase this to characterise the meaning of a thing on the web by the company of things linked to it. In order for such meaning to be representative, though, we have to analyse the 'company' across as much content as possible. To do so, we employ an approach utilising simple, yet universal and powerful tensor-based representation of distributional semantics proposed in [3]. We adapt it to the Semantic Web specifics and show how one can execute rules on the top of it, which effectively leads to a smooth combination of the bottom-up (distributional) and top-down (symbolic) approaches to the representation of meaning. Apart of that, we dedicate a substantial a part of the paper (Section 4) to an experimental application of our approach to automated consolidation of knowledge in life sciences. We conclude the paper in Section 5.

2 Related Work

We define emergent meaning of Semantic Web expressions using tensors to elucidate various distributional effects, which stems from the comprehensive approach in [3]. However, we extended this approach with a symbolic (rule-based) layer in order to combine it with the top-down Semantic Web principles. A tensor-based representation of the Semantic Web data was presented for instance in [4], which,

however, focuses mostly on ranking and decomposition, not on providing generic means for an analysis of various bottom-up semantics phenomena. Approaches to induction of implicit patterns or schemata from data are also related to our work (particular examples include [5] and [6] in the fields of databases and Semantic Web, respectively). Yet these approaches are usually focused on rather limited sets of problems (e.g., query optimisation or concept induction) and thus are not as comprehensive and theoretically uniform as our framework. The NELL project [7] aims at incremental and continuous induction of triple patterns from the web data, which is a goal very similar to ours. The main differences are that NELL needs manually provided seeds of knowledge and a slight supervision in the form of pruning. Also, the type of extracted patterns is limited to instances of fixed generic relations in NELL, whereas we allow for bottom-up inference of rather varied and dynamic set of phenomena. Works like [8] or [9] deal with emergent semantics, but they mostly investigate how to gather the semantics (from ontologies or simple peer-to-peer interactions in heterogeneous data systems). Less attention is paid to how the semantics can be uniformly represented and utilised later on. Finally, our experiment in knowledge consolidation is closely related to ontology matching [10] and life science data integration [11]. Most of the ontology matching algorithms are designed to operate at the schema level, though, and not at the data level that is most pertinent to our work. Regarding extant methods for knowledge integration in life sciences, majority of them uses quite a limited set of specifically tuned lexical or structural similarities. Thus our approach can provide for a more adaptive and empirically driven data-based integration in this context.

3 Distributional Web Semantics

The proposed distributional web semantics framework has two major layers – the bottom-up and top-down one. The former caters for the implicit meaning, while the latter allows for adding more value to the bottom-up analysis by utilising the current Semantic Web resources (e.g., RDF Schema or ontologies). A general way of using the framework follows this pipeline: (1) convert a set of simple RDF documents into the internal distributional representation; (2) extract interesting patterns from it; (3) make use of extant top-down semantic resources to materialise more implicit knowledge by means of inference (optional); (4) utilise the results to improve the quality of the initial RDF data set. The last step can consist of exporting the distributional patterns as RDF statements to be added to the input data (e.g., as links between the entities or properties found to be similar). Alternatively, one can present the patterns directly to users along the original data set to facilitate its machine-aided augmentation.

3.1 Bottom-Up Layer

Source Representation The basic structure of the bottom-up layer is a so called source (or graph) representation $\mathbf{G}$, which captures the co-occurrence of

things (i.e., subjects and objects) within relations (i.e., predicates) across a set of documents (i.e., RDF graphs). Let A_l, A_r be sets representing left and right arguments of binary co-occurrence relationships (i.e., statements), and L the types of the relationships. A_l, A_r, L correspond to sets of RDF subjects, objects and predicates, respectively. Furthermore, let P be a set representing provenances of particular relationships (i.e., graph names). We define the source representation as a 4-ary labeled tensor $\mathbf{G} \in \mathbb{R}^{|A_l| \times |L| \times |A_r| \times |P|}$. It is a four-dimensional array structure indexed by subjects, predicates, objects and provenances, with values reflecting a frequency or weight of statements in the context of particular provenance sources (0 if a statement does not occur in a source). For instance, if a statement (a_l, l, a_r) occurs k-times in a data source d (a single graph or a set of graphs in general), then the element $g_{a_l,l,a_r,d}$ of $\mathbf{G}$ will be set to k to reflect it. More details are illustrated in the following example.

Example 1. Let us consider 7 statements (acquired from biomedical texts):

> (*protein domain* , *different*, *protein*), (*protein domain*, *type*, *domain*), (*gene*, *different*, *protein*), (*internal tandem duplications*, *type*, *mutations*), (*internal tandem duplications*, *in*, *juxtamembrane*), (*internal tandem duplications*, *in*, *extracelullar domains*), (*protein domain*, *type*, *domain*)

with provenances $D_1, D_1, D_2, D_3, D_3, D_3, D_4$, respectively. The source representation (using statement occurrence frequencies as values) is:

$s \in A_l$	$p \in L$	$o \in A_r$	$d \in P$	$g_{s,p,o,d}$
protein domain	*different*	*protein*	D_1	1
protein domain	*type*	*domain*	D_1	1
gene	*different*	*protein*	D_2	1
internal tandem duplications	*type*	*mutations*	D_3	1
internal tandem duplications	*in*	*juxtamembrane*	D_3	1
internal tandem duplications	*in*	*extracelullar domains*	D_3	1
protein domain	*type*	*domain*	D_4	1

We omit all zero values and use the tabular notation as a convenient and concise representation of a 4-dimensional tensor, with the three first columns for indices and the fourth one for the corresponding value.

Corpus Representation The source tensor is merely a low-level data representation preserving the association of statements with their provenance contexts. Before allowing for actual distributional analysis, the data have to be transformed into a more compact structure $\mathbf{C}$ called corpus representation. $\mathbf{C} \in \mathbb{R}^{|A_l| \times |L| \times |A_r|}$ is a ternary (three-dimensional) labeled tensor, devised according to [3] in order to provide for a universal and compact distributional representation for the proposed bottom-up web semantics framework. A corpus $\mathbf{C}$ can be constructed from a source representation $\mathbf{G}$ using functions $a : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, w : P \to \mathbb{R}, f : A_l \times L \times A_r \to \mathbb{R}$. For each $\mathbf{C}$ element $c_{s,p,o}$, $c_{s,p,o} = a(\sum_{d \in P} w(d) g_{s,p,o,d}, h(s,p,o))$, where $g_{s,p,o,d}$ is an element of the source tensor $\mathbf{G}$ and the a, f, w functions act as follows: (1) w assigns a relevance degree to each source; (2) f reflects the relevance of the statement elements (e.g., a mutual information score of the subject and object within the sources); (3) a aggregates the result of the w, f functions' application. This way of constructing the elements of the corpus tensor from the low-level source representation

essentially aggregates the occurrences of statements within the input data, reflecting also two important things – the relevance of particular sources (via the w function), and the relevance of the statements themselves (via the f function). The specific implementation of the functions is left to applications – possible examples include (but are not limited to) ranking (both at the statement and document level) or statistical analysis of the statements within the input data. In Section 4.1, we provide a detailed description of a particular source-to-corpus conversion we used in the evaluation experiment.

Example 2. A corpus corresponding to the source tensor from Example 1 can be represented (again in a tabular notation) as given below. The w values were 1 for all sources and a, f aggregated the source values using relative frequency (in a data set containing 7 statements).

$s \in A_l$	$p \in L$	$o \in A_r$	$c_{s,p,o}$
protein domain	$different$	*protein*	1/7
protein domain	*type*	*domain*	2/7
gene	$different$	*protein*	1/7
internal tandem duplications	*type*	*mutations*	1/7
internal tandem duplications	*in*	*juxtamembrane*	1/7
internal tandem duplications	*in*	*extracelullar domains*	1/7

Corpus Perspectives The elegance and power of the corpus representation lays in its compactness and universality that, however, yields for many diverse possibilities of the underlying data analysis. The analysis are performed using a process of so called matricisation of the corpus tensor $\mathbf{C}$. Essentially, matricisation is a process of representing a higher-order tensor using a 2-dimensional matrix perspective. This is done by fixing one tensor index as one matrix dimension and generating all possible combinations of the other tensor indices within the remaining matrix dimension. In the following we illustrate the process on the simple corpus tensor from Example 2. Detailed description of matricisation and related tensor algebra references can be found in [3].

Example 3. When fixing the subjects (A_l set members) of the corpus tensor from Example 2, one will get the following matricised perspective (the rows and columns with all values equal to zero are omitted here and in the following examples):

$s/\langle p, o\rangle$	$\langle d, p\rangle$	$\langle t, dm\rangle$	$\langle t, m\rangle$	$\langle i, j\rangle$	$\langle i, e\rangle$
protein domain	1/7	2/7	0	0	0
gene	1/7	0	0	0	0
internal tandem duplications	0	0	1/7	1/7	1/7

.

The abbreviations d, p, t, dm, m, i, j, e stand for $different$, *protein*, *type*, *domain*, *mutations*, *in*, *juxtamembrane*, *extracellular domains*. One can clearly see that the transformation is lossless, as the original tensor can be easily reconstructed from the matrix by appropriate re-grouping of the indices.

The corpus tensor matricisations correspond to vector spaces consisting of elements defined by particular rows of the matrix perspectives. Each of the vectors has a name (the corresponding matrix row index) and a set of features (the matrix column indices). The features represent the distributional

attributes of the entity associated with the vector's name – the contexts aggregated across the whole corpus. Thus by comparing the vectors, one essentially compares the meaning of the corresponding entities emergently defined by the underlying data. For exploring the matricised perspectives, one can uniformly use the linear algebra methods that have been successfully applied to vector space analysis tasks for the last couple of decades. Large feature spaces can be reliably reduced to a couple of hundreds of the most significant indices by techniques like singular value decomposition or random indexing (see `http://en.wikipedia.org/wiki/Dimension_reduction` for details). Vectors can be compared in a well-founded manner by various metrics or by the cosine similarity (see `http://en.wikipedia.org/wiki/Distance` or `http://en.wikipedia.org/wiki/Cosine_similarity`, respectively). This way matrix perspectives can be combined with vector space analysis techniques in order to investigate a wide range of semantic phenomena related to synonymy, clustering, ambiguity resolution, taxonomy detection or analogy discovery. In this introductory paper, we focus only on clustering of similar entities (subjects and/or objects) and properties. The following example explains how to perform these particular types of analysis.

Example 4. Let us add two more matrix perspectives to the $s/\langle p, o\rangle$ one provided in Example 3. The first one represents the distributional features of objects (based on the contexts of predicates and subjects they tend to co-occur with in the corpus):

$o/\langle p, s\rangle$	$\langle d, pd\rangle$	$\langle t, pd\rangle$	$\langle d, g\rangle$	$\langle t, itd\rangle$	$\langle i, itd\rangle$
$protein$	1/7	0	1/7	0	0
$domain$	0	2/7	0	0	0
$mutations$	0	0	0	1/7	0
$juxtamembrane$	0	0	0	0	1/7
$extracellular\ domains$	0	0	0	0	1/7

.

d, pd, t, g, itd, i stand for *different, protein domain, type, gene, internal tandem duplications, in.* Similarly, the second perspective represents the distributional features of properties:

$p/\langle s, o\rangle$	$\langle pd, p\rangle$	$\langle pd, d\rangle$	$\langle g, p\rangle$	$\langle itd, m\rangle$	$\langle itd, j\rangle$	$\langle itd, ed\rangle$
$different$	1/7	0	1/7	0	0	0
$type$	0	2/7	0	1/7	0	0
in	0	0	0	0	1/7	1/7

.

$itd, pd, p, d, g, m, j, ed$ stand for *internal tandem duplications, protein domain, protein, domain, gene, mutations, juxtamembrane, extracellular domains.*

The vector spaces induced by the matrix perspectives $s/\langle p, o\rangle$ and $o/\langle p, s\rangle$ can be used for finding similar entities by comparing their corresponding vectors. Using the cosine vector similarity, one finds that $sim_{s/\langle p,o\rangle}(protein\ domain, gene) = \frac{\frac{1}{7}\frac{1}{7}}{\sqrt{(\frac{1}{7})^2+(\frac{2}{7})^2}\sqrt{(\frac{1}{7})^2}} \doteq 0.2972$ and $sim_{o/\langle p,s\rangle}(juxtamembrane, extracel\text{-}lular\ domains) = \frac{\frac{1}{7}\frac{1}{7}}{\sqrt{(\frac{1}{7})^2}\sqrt{(\frac{1}{7})^2}} = 1$. These are the only non-zero similarities among the subject and object entities present in the corpus. As for the predicates, all of them have a zero similarity. This quite directly corresponds to the intuition a human observer can get from the data represented by the initial statements from Example 1. Protein domains and genes seem to be different from proteins, yet protein domain is a type of domain

and gene is not, therefore they share some similarities but are not completely equal according to the data. Juxtamembranes and extracellular domains are both places where internal tandem duplications can occur, and no other information is available, so they can be deemed equal until more data comes. Among the particular predicates, no patterns as clear as for the entities can be observed, therefore they can be considered rather dissimilar given the current data.

3.2 Top-Down Layer

A significant portion of the expressive Semantic Web standards (RDFS, OWL) and widely used extensions (such as N3, cf. `http://www.w3.org/DesignIssues/Notation3.html`) can be expressed by conjunctive rules (see `http://www.w3.org/TR/rdf-mt/`, `http://www.w3.org/TR/owl2-profiles/` or [12]). To allow for a seamless combination of this top-down layer of the Semantic Web with the bottom-up principles introduced in the previous section, we propose a straightforward adaptation of state of the art rule-based reasoning methods.

Conjunctive rules can be described as follows in the 'language' of the bottom-up semantics. Let $\mathcal{S} = \mathbb{R}^{|A_l \cup V| \times |L \cup V| \times |A_r \cup V|}$ be a set of corpus tensors with their index domains (A_l, L, A_r) augmented by a set of variables V. Then $(\mathbf{L}, \mathbf{R}, w)$, where $\mathbf{L}, \mathbf{R} \in \mathcal{S}, w \in \mathbb{R}$, is a rule with an antecedent $\mathbf{L}$, a consequent $\mathbf{R}$ and a weight w. The values of the rule tensors are intended to represent the structure of the rule statements – a non-zero value reflects the presence of a statement consisting of the corresponding indices in the rule. However, the antecedent tensor values can also specify the weights of the relationship instances to be matched and thus facilitate uncertain rule pattern matching. The weights can be used to set relative importance of rules. This is especially useful when combining rules from rule sets of variable relevance – one can assign higher weights to rule coming from more reliable resources and the other way around. We assume the weights to be set externally – if this is not the case, they are assumed to be 1 by default.

Example 5. An RDFS entailment rule for transitivity can be stated in N3 as: `{?x rdfs:subClassOf ?y . ?y rdfs:subClassOf ?z } => {?x rdfs:subClassOf ?z }.` The rule is transformed to the tensor form as:

$s \in A_l \cup V$	$p \in L \cup V$	$o \in A_r \cup V$	$l_{s,p,o}$
$?x$	$rdfs:subClassOf$	$?y$	1
$?y$	$rdfs:subClassOf$	$?z$	1

,

$s \in A_l \cup V$	$p \in L \cup V$	$o \in A_r \cup V$	$r_{s,p,o}$
$?x$	$rdfs:subClassOf$	$?z$	1

, 1).

Rules can be applied to a corpus by means of Algorithm 1. The particular rule-based reasoning method we currently use is a modified version of the efficient RETE algorithm for binary predicates [13]. The $conditionTrees()$ function in Algorithm 1 generates a set of trees of antecedent conditions from a rule set $\mathcal{R}$.

Example 6. For instance, let us imagine the following rule set (described in N3 again): R_1 : `{?x rdfs:subClassOf ?y . ?y rdfs:subClassOf ?z } => {?x rdfs:subClassOf ?z}.` R_2 :`{?x rdfs:subClassOf ?y . ?z rdf:type ?x} => {?z rdf:type ?y}.`

Algorithm 1 Rule Evaluation

1: $RESULTS \leftarrow \emptyset$
2: $FOREST \leftarrow conditionTrees(\mathcal{R})$
3: **for** $T \in FOREST$ **do**
4: **for** $(I, \mathbf{R}, w) \in matches(T)$ **do**
5: $\mathbf{R}' \leftarrow w \cdot materialise(I, \mathbf{R})$
6: $RESULTS \leftarrow RESULTS \cup \mathbf{R}'$
7: **end for**
8: **end for**
9: **return** $\sum_{\mathbf{X} \in RESULTS} \mathbf{X}$

For simplicity, we assume the weights of the rules R_1, R_2 to be 1.0. Given this rule set, the *conditionTrees*() function returns a single tree with a root condition `?x rdfs:subClassOf ?y` and the `?y rdfs:subClassOf ?z`, `?z rdf:type ?x` conditions as the root's children. The tree leafs (i.e., children of the root's children) then point to the consequents and weights of the rules R_1, R_2, respectively.

The rule condition forest allows for optimised incremental generation of all possible corpus instance assignments to the variables in the rule conditions – each condition is being evaluated only once even if it occurs in multiple rules. The generation of instance assignments for particular condition variables is realised by the function *matches*() in Algorithm 1. It produces tuples $(I, \mathbf{R}, w)$, where I is an assignment of instances to the antecedent variables along a particular root-leaf path in the given tree T. $\mathbf{R}, w$ are then the rule consequent and weight in the leaf of the corresponding instance assignment path.

The function *materialise*() takes the computed instance assignment I and applies it to the consequent $\mathbf{R}$. The values of the materialised consequent tensor $\mathbf{R}'$ are computed as $r_{s,p,o} = \top\{c_{i_1,i_2,i_3} | (i_1, i_2, i_3) \in I\}$ for each (s, p, o) element of the consequent that has a non-zero value in the original $\mathbf{R}$ tensor. The c_{i_1,i_2,i_3} elements of the tensor $\mathbf{C}$ (the corpus representation, i.e., knowledge base) correspond to all statements in the instantiated rule conditions along the assignment path I. Finally, the $\top$ operation is an application of a fuzzy conjunction (t-norm, cf. `http://en.wikipedia.org/wiki/T-norm`) to a set of values[1]. The result of Algorithm 1 is a sum of all the tensors resulting from the particular consequent materialisations weighted by the corresponding rule weights.

Example 7. To exemplify an iterative rule materialisation (knowledge base closure), let us add two more elements to the corpus from Example 2 (the weights are purely illustrative):

A_l	L	A_r	value
domain	$rdfs:subClassOf$	*molecular structure*	2/9
molecular structure	$rdfs:subClassOf$	*building block*	1/9

[1] Note that although the minimum t-norm, $t(a, b) = min(a, b)$, can be applied to any positive values in the corpus representation tensors with the intuitively expected (fuzzy-conjunctive) semantics, any other t-norm, such as the product one, $t(a, b) = ab$, would lead to rather meaningless results if the tensor values were not normalised to the $[0, 1]$ interval first.

If we assume that the `type` relation from the previous examples is equivalent to the `rdf:type` relation from the rule R_2 in Example 6, we can apply the R_1, R_2 rules to the extended corpus representation with the following results. After assigning instances to the antecedent variables, the only instance path leading towards R_1 in the condition tree consists of the statements `domain rdfs:subClassOf molecular structure` and `molecular structure rdfs:subClassOf building block`. The R_2 branch generates four possible instance paths. The root can have two values: `domain rdfs:subClassOf molecular structure`, `molecular structure rdfs:subClassOf building block`. Similarly for the child – there are two statements in the corpus that fit the corresponding condition: `protein domain rdf:type domain` and `internal tandem duplications rdf:type mutations`. When using the minimum t-norm we can enrich the knowledge base by the following materialised consequents:

$s \in A_l$	$p \in L$	$o \in A_r$	$r_{s,p,o}$
domain	$rdfs:subClassOf$	*building block*	1/9
protein domain	$rdf:type$	*molecular structure*	2/9

If we apply Algorithm 1 again, we get one more new statement:

$s \in A_l$	$p \in L$	$o \in A_r$	$r_{s,p,o}$
protein domain	$rdf:type$	*building block*	2/9

After that the corpus representation already remains stable (its closure has been computed), as no further application of the rules produces new results.

4 Evaluation

In the evaluation, we addressed life sciences, a domain where the information overload is now more painful than ever and where efficient data/knowledge integration can bring a lot of benefit [11]. Specifically, we looked into knowledge consolidation, by which we mean—at the abstract level—grouping of possibly isolated, yet related simple facts into more general chunks of knowledge with similar meaning. Drilling down to a more concrete level of the actual experiments, we applied the framework proposed in this paper to clustering of entities (i.e., subjects and objects) and relations based on their distributional features within a corpus of input resources. We considered two types of inputs – existing linked data sets and statements extracted from texts associated with the linked data sets. Details on the data, experiments, evaluation methods and results are provided in the corresponding sections below.

4.1 Method

Data The first type of data we used were four RDF documents (parts of the Linked Open Data cloud) that were converted into RDF from manually curated life science databases and served on the D2R web site (`http://www4.wiwiss.fu-berlin.de/`). To keep the data set focused, we chose resources dealing with drugs and diseases: Dailymed, Diseasome, Drugbank and Sider (see `http://goo.gl/c0Dqo`, `http://goo.gl/sbq8E`, `http://goo.gl/ydMSD` and `http://goo.gl/LgmlF`, respectively). This data set is referred to by the LD identifier in the rest

of the paper. We pre-processed the data as follows. Most importantly, we converted the identifiers of entities to their human-readable names to facilitate the evaluation. Also, we added new statements for each explicitly defined synonym in the LD data set by "mirroring" the statements of the descriptions associated with the corresponding preferred term. More technical details are available at `http://goo.gl/38bGK` (an archive containing all the data, source code and additional descriptions relevant to the paper).

The second data set we used was generated from the textual content of the LD documents, which contain many properties with string literal objects representing natural language (English) definitions and detailed descriptions of the entries (e.g., `drugbank:pharmacology` describes the biochemical mechanism of drug functions). We extracted the text from all such properties, cleaned it up (removing spurious HTML mark-up and irregularities in the sentence segmentation) and applied a simple NLP relation extraction pipeline on it, producing a data set of extracted statements (XD in the following text). In the extraction pipeline we first split the text into sentences and then associated each word in a sentence with a corresponding part-of-speech tag. The tagged sentences were shallow-parsed into a tree structure with annotated NPs (noun phrases). These trees were then used to generate statements particular statements as follows. From any NP_1 $[verb|preposition]^+$ NP_2 sequence in the parsed tree, we created subject from NP_1, object from NP_2 and predicate from the intermediate verb or prepositional phrase. Additional statements were generated by decomposing compound noun phrases. More details and examples are out of scope here, but we made them available for interested readers as a part of the data package provided at `http://goo.gl/38bGK`.

Concerning the size of the experimental data, the linked data sets contained ca. 630 thousand triples, 126 properties and around 270 thousands of simple entities (i.e., either subjects or objects) corresponding to almost 150 thousands of unique identifiers (i.e., preferred labels). The size of the extracted data set was around 3/4 of the linked data one, however, the number of extracted properties was much higher – almost 35 thousand. Apart of the LD, XD data sets, we also prepared their LD$^-$, XD$^-$ alternatives, where we just 'flattened' all the different properties to uniform links. We did so to investigate the influence the multiple property types have on the distributional web semantics features within the experiments.

Knowledge Consolidation Before performing the knowledge consolidation, we had to incorporate the RDF data (the LD, XD sets) into the framework introduced in Section 3, i.e., to populate the graph and source representation tensors $\mathbf{G}, \mathbf{C}$ (separate tensors for each of the LD, XD, LD$^-$, XD$^-$ data sets). The $\mathbf{G}$ indices were filled by the lexical elements of triples and by the corresponding source graph identifiers (there were five provenance graphs – one for each of the four linked data documents and one for the big graph of extracted statements). The $\mathbf{G}$ values were set to 1 for all elements $g_{s,p,o,d}$ such that the statement (s, p, o) occurred in the graph d; all other values were 0. To get the $\mathbf{C}$ tensor values $c_{s,p,o}$, we multiplied the frequency of the (s, p, o) triples (i.e., $\sum_{d \in P} g_{s,p,o,d}$)

by the point-wise mutual information score of the (s, o) tuple (see `http://en.wikipedia.org/wiki/Pointwise_mutual_information` for details on the mutual information score theory and applications). This method is widely used for assigning empirical weights to distributional semantics representations [3], we only slightly adapted it to the case of our "triple corpora" by using the frequencies of triple elements and triples themselves. As we were incorporating triples from documents with equal relevance, we did not use any specific provenance weights in the $\mathbf{C}$ tensor computation. After the population of the corpus tensor, we used its $s/\langle p, o\rangle$, $o/\langle p, s\rangle$ perspectives for generating similar entities and the $p/\langle s, o\rangle$ perspective for similar properties, proceeding exactly as described in Example 4. A cluster of size x related to a vector $\mathbf{u}$ in a perspective π was generated as a set of up to x most similar vectors $\mathbf{v}$ such that $sim_\pi(\mathbf{u}, \mathbf{v}) > 0$. Our implementation of the large scale tensor/matrix representation and analysis is open source and available as a part of the data package provided at `http://goo.gl/38bGK`. Note that one might also employ rules in the knowledge consolidation experiments, for instance to materialise more implicit statements providing additional features for the distributional analysis. However, a comprehensive evaluation of such a combined approach does not fit into the scope of this paper, therefore we will elaborate on it in a separate technical report.

To evaluate the entity consolidation, we employed a gold standard – MeSH (see `http://www.nlm.nih.gov/mesh/`), a freely available controlled vocabulary and thesaurus for life sciences. MeSH is manually designed and covers a lot of disease, gene and drug terms, therefore the groups of related things within its taxonomical structure are a good reference comparison for artificially generated clusters of entities from the same domain[2]. To the best of our knowledge, no similar applicable gold standard that would cover our property consolidation data sets exists, thus we had to resort to manual assessment of the corresponding results. As a baseline, we used randomly generated clusters of entities and properties. Other baseline methods are possible, such as various ontology matching techniques [10]. However, these methods are designed rather for 'schema-level' matching between two semantic resources. Their application to the 'data-level' consolidation of many statements possibly contained in a single resource is a research question in its own right, which we leave for future work.

Evaluation Metrics For the entity clustering evaluation, we first need a 'gold standard' similarity between two terms, based on their paths in the MeSH taxonomy[3]. Every MeSH entry (and its synonyms) are associated with one or more

[2] We considered, e.g., GO, as well as vaccine and disease ontologies from the OBO dataset for the gold standard. Unfortunately, either the coverage of the ontologies w.r.t. the experimental dataset was worse than for MeSH, or the definition of 'gold standard' similarities was trickier (requiring possibly substantial further research) due to additional relations and/or rather complex (OWL) semantics. Thus, for the time being, we chose to focus on something simpler, but representative and already convincing.

[3] This essentially edge-based approach is motivated by the similarity measures commonly used in the context of life science knowledge bases [14]. A possible alternative would be a node-based similarity utilising the information content measure (also

tree codes. These take form of alphanumeric tree level identifiers divided by dots that determine the position of particular entries in the MeSH trees. The path from the root (most generic term in a category) to a given entry corresponds to its tree code read from left to right. For instance, *abdominal wall* and *groin* have MeSH tree codes *A01.047.050* and *A01.047.365*, which means they have a path of length 2 in common from their tree's root (going from *Body Regions* through *Abdomen* with the respective tree codes *A01, A01.047*). The length of the path shared by two terms can be used as a simple and naturally defined similarity measure – the bigger the relative portion of the MeSH root path shared by two terms, the closer—i.e., more similar—they are in the MeSH hierarchy.

More formally, we can define a MeSH similarity measure s_M between two terms s, t as follows. If s and t do not share any node in their tree paths, $s_M(s,t) = 0$. If any of the s, t tree paths subsumes the other one, $s_M(s,t) = 1$. In the remaining cases, $s_M(s,t) = k^{max(|p(s)|,|p(t)|)-|mcp(s,t)|}$, where $k \in (0,1)$ is a coefficient, $p(x)$ refers to the tree path of a term x and $mcp(s,t)$ is a maximum common path shared between the tree paths of s, t. For our experiment, we chose coefficient $k = 0.9$, which provides for clearly visible but not too abrupt changes in the similarity values. Note that the particular choice of k is rather cosmetic as it does not influence the descriptive power of the results, it only changes the absolute values of the similarity.

To assess the clusters computed in the entity consolidation experiments, we defined the overlap (o) and quality (q) evaluation measures: $o(C) = \frac{|\{t|t \in M \wedge t \in C\}|}{|C|}$, $q(C) = \frac{\sum_{(s,t) \in comb_2(C_M)} s_M(s,t)}{|comb_2(C_M)|}$, where C is the cluster (a set of terms) being measured, M is a set of all MeSH terms, $C_M = \{t|t \in M \wedge t \in C\}$ are all terms from C that are in MeSH as well, $comb_2(C_M)$ selects all combinations of two different terms from C_M, and, finally, s_M is the MeSH term similarity. If $C_M = \emptyset$, $q(C) = 0$ by definition. The overlap is an indication of how many terms from MeSH are contained in a cluster, while the quality is the actual evaluation measure that tells us how good the part of the cluster covered by the gold standard is (i.e., how close it is to the structure of the manually designed MeSH thesaurus). The quality is computed as an average similarity of all possible combinations of term pairs in a cluster that are contained in MeSH. Such a measure may seem to be a little restrictive when the MeSH-cluster overlap is low. However, the low overlap is not as much caused by the noise in the clusters as by insufficient coverage of the gold standard itself, which is quite a common problem of gold standards in as dynamic and large domains as life sciences [11].

Since we lack a proper gold standard for the property consolidation experiment, the corresponding evaluation measures will inevitably be a slightly less

discussed in detail in [14]). However, the computation of the information content depends on a clear distinction between classes (or concepts) and instances (or terms) in the data set. In our case, this is rather difficult – an instance term can become a class term as soon as it becomes a type of another term in an extracted statement. Therefore an application of a node-based (or a hybrid node-edge) similarity would require additional investigations that unfortunately do not fit the rather limited scope of this paper.

solid than the ones for entity clustering. We base them on human assessment of two factors – an adequacy (aq) and accuracy (ac) of property clusters. Given a property cluster C, $ac = \frac{|C|-|N|}{|C|}$, $aq = \frac{|R|}{|C|-|N|}$, where N is a set of properties deemed as noise by a human evaluator, and R is a set of properties considered to be relevant to the seed property the cluster was generated from. For the manual evaluation of the property consolidation experiment, we used two human experts (one bioinformatician and one clinical researcher). They both had to agree on determining whether properties are not noise and whether they are relevant. For the opposite decisions, a single vote only was enough to mark the corresponding property as a true negative result (thus making the evaluation rather pessimistic in order to reduce the likelihood of bias).

In both entity and property consolidation experiments, we randomly selected 10 batches of 10 terms that served as cluster seeds for each evaluated cluster size, and computed the arithmetic means of the metrics of all the clusters. Thus we made sure that the results closely approximated the whole data set (for the automatic evaluation with gold standard, at least 75% of the particular results in all the selected batches were differing from the mean value by less than 5%, although some fluctuations were present in the manual evaluation). We tested several different sizes of the generated clusters – $10^{<}, 10, 25, 50, 100, 250, 500$. The $10^{<}$ clusters contained terms related to the seed term with a similarity of at least 0.75 (no such cluster was larger than 10 when abstracting from synonyms). Most interesting clusters were of size up to 50, bigger sizes already converged to the random baseline.

4.2 Results and Discussion

The evaluation results are summarised in Table 1. The first column represents

cl. size	$10^{<}$		10		25		50		100		250		500	
metric	*o*	*q*	*o*	*q*	*o*	*q*	*o*	*q*	*o*	*q*	*o*	*q*	*o*	*q*
EC-LD	0.021	0.071	0.048	0.103	0.024	0.070	0.017	0.064	0.019	0.052	0.031	0.071	0.019	0.062
EC-LD^{-}	0.075	0.248	0.055	0.176	0.060	0.285	0.066	0.258	0.079	0.242	0.086	0.237	0.084	0.199
EC-XD	0.021	0.045	0.016	0.047	0.022	0.042	0.030	0.081	0.029	0.064	0.023	0.050	0.037	0.073
EC-XD^{-}	0.053	0.127	0.038	0.109	0.049	0.121	0.046	0.109	0.067	0.127	0.072	0.130	0.091	0.092
EC-BL	0.011	0.000	0.020	0.000	0.040	0.000	0.044	0.110	0.064	0.130	0.067	0.118	0.066	0.119
metric	*aq*	*ac*	*aq*	*ac*	*aq*	*ac*	*aq*	*ac*	*aq*	*ac*	*aq*	*ac*	*aq*	*ac*
PC-LD	0.875	1.000	0.603	1.000	0.578	1.000	0.596	1.000	N/A	N/A	N/A	N/A	N/A	N/A
PC-BL-LD	0.134	1.000	0.140	1.000	0.048	1.000	0.027	1.000	N/A	N/A	N/A	N/A	N/A	N/A
PC-XD	0.417	0.448	0.593	0.550	0.395	0.429	0.450	0.589	N/A	N/A	N/A	N/A	N/A	N/A
PC-BL-XD	0.016	0.497	0.027	0.450	0.017	0.523	0.024	0.511	N/A	N/A	N/A	N/A	N/A	N/A

Table 1. Results summary

labels of the experimental data sets. The EC, PC parts of the labels indicate the entity and property consolidation experiments. XD and LD refer to the usage of the extracted and linked data sets. The $^{-}$ superscripts refer to flattened, "property-less" data sets. Finally, EC-BL, PC-BL-LD, PC-BL-XD refer to the

particular random baseline experiments. The columns of Table 1 represent the evaluation measures per each tested cluster size.

Firstly we will discuss the results of property clustering. To reduce the already quite substantial workload of the human evaluators, we considered only clusters of size up to 50. The accuracy of the LD batch was obviously 1, since the properties were manually defined there. The adequacy of clustering was best (0.875) for small, crisp LD clusters (with a rather strict similarity threshold of 0.75), while for bigger clusters without a similarity threshold restriction, it was decreasing, yet still significantly better than the baseline. For the extracted data set (XD), roughly one half of extracted properties was deemed to be accurate. Out of these, around 46.4% in average were adequate members of the analysed property clusters, which is quite promising, as it would allow for reduction of the space of about $35,000$ extracted properties to several hundreds with an error rate around 50%. This may not be enough for a truly industry-strength solution, but it could already be useful in prototype applications if extended by result filtering (e.g., ranking). Examples of interesting property clusters we generated are: $C_1 = \{$*secreted_in, excreted_in, appear_in, detected_in, accounted_for_in, produced_in, eliminated_in*$\}$, $C_2 = \{$*from, following_from, gathered_from*$\}$, $C_3 = \{$*increase, increased_by, diminish*$\}$. C_1 appears to be related to production/consumption of substances in organs, C_2 to origin of substances in location and C_3 to quantity change. More examples are available in the data package at `http://goo.gl/38bGK`.

To discuss the entity clustering results, let us have a look at Figure 1, which shows the dependency of the quality (q) measure on the cluster sizes for all the evaluated data sets. The dotted green line (circle markers) represents the baseline (EC-BL), while the red and black lines (square/plus and diamond/cross markers) are for the extracted and linked open data sets, respectively (EC-XD/EX-XD$^-$, EC-LD/EC-LD$^-$). The solid lines are for original data sets (with properties), whereas the dashed lines indicate resources with properties reduced to mere links. One can immediately see that the results of entity consolidation are significantly better in terms of quality than the baseline for clusters of size up to 25. This holds for all evaluated data sets and thus demonstrates a clear contribution of our approach. This is, we believe, not the most interesting thing, though. Quite surprisingly, the data sets flattened to mere links between entities (the dotted lines) produce much better results than the original resources with multiple property types. This is especially the case of flattened linked data resources (the dashed black line), which perform better than the baseline for all cluster sizes. Another counterintuitive finding is that the flattened automatically extracted resources (red dashes) perform better than the manually created linked data sets (without flattened properties). For clusters of size 50 and bigger, the flattened extracted batch oscillates around the random baseline, while both batches with actual properties are consistently worse.

It would be easy to say that the reason for such results is that the properties in triples do not have any semantics that can be empirically useful. This would be quite a controversial claim in the Semantic Web context, though, and we

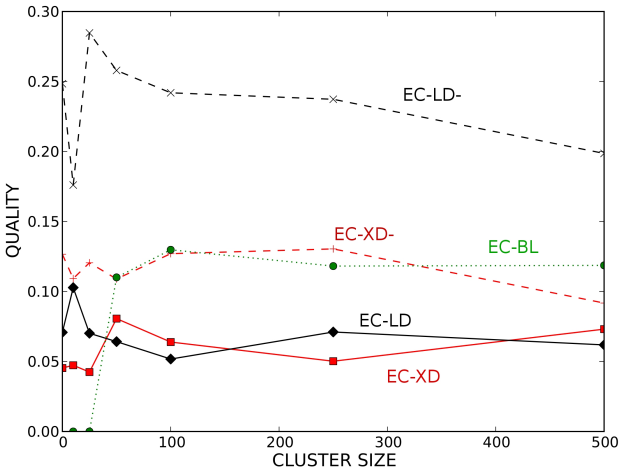

Fig. 1. Dependency of the cluster quality their sizes

believe that it is most likely false, as the properties have been proven useful in countless other applications already. Alternative explanation could be that particular authors of the original linked data resources used the properties in a rather haphazard way when contributing to the knowledge base, thus introducing noise at a larger scale. This might partly be a culprit of the observed results, but there is yet another, perhaps more intriguing and plausible hypothesis: what if the empirical similarities induced by the full and flattened data are actually different? The gold standard similarity imposed by the MeSH thesaurus may be directly related to its taxonomy structure. The flattened resources may produce a related, rather simple 'subsumption' type of distributional similarity, while the resources with multiple property types can give rise to a more complex 'structural' similarity. This could be the reason for a better fit of the flattened data to the gold standard. Also, it could explain the poor (i.e., worse than random) performance of the full-fledged data sets for larger cluster sizes. In these cases, the flattened resources may be producing bigger clusters of more general and more specific (but still related) terms, whereas the other type of similarity just increases the noise by adding more and more specific and mutually unrelated structural sub-clusters. Experimenting with alternative similarity measures for the comparison of the results with the gold standard should help to clarify these rather surprising results. Apart of the hypothetical explanations mentioned so far, also the actual method of computing the corpus representation values (see Section 3.1) may play a significant role here. Whatever the actual reasons for the obtained results are, we believe that a further investigation of the suggested hypotheses could lead to interesting findings about more fundamental principles of the web semantics than investigated in this introductory paper.

5 Conclusion and Future Work

We have proposed to complement the currently prevalent top-down approaches to web semantics by an additional distributional layer. This layer allows for so far unexplored representation and analysis of bottom-up phenomena emerging from the Semantic Web resources. We demonstrated the usefulness of our framework by applying it to an experiment in consolidation of life science knowledge. The results showed promising potential of our approach and, in addition, revealed unexpected findings that are inspiring for futher research.

Our next plans include an experiment with a full-fledged combination of the top-down and bottom-up semantics (i.e., with rule-based materialisations in the loop). We will also explore a far wider range of the distributional semantics phenomena within various practical applications (e.g., automated thesaurus construction, rule learning or discovery of analogies). Then we want to engage in a continuous collaboration with sample users to assess qualitative aspects and practical value of tools based on our work. Finally, we want to ensure web-scale applicability of our approach by its distributed and parallel implementations.

References

1. de Saussure, F.: Course in General Linguistics. Open Court, Illinois (1983)
2. Firth, J.: A synopsis of linguistic theory 1930-1955. Studies in Ling. Anal. (1957)
3. Baroni, M., Lenci, A.: Distributional memory: A general framework for corpus-based semantics. Computational Linguistics (2010)
4. Franz, T., Schultz, A., Sizov, S., Staab, S.: Triplerank: Ranking semantic web data by tensor decomposition. In: IWSC, Springer (2009)
5. Goldman, R., Widom, J.: DataGuides: Enabling query formulation and optimization in semistructured databases. In: VLDB, Morgan Kaufmann (1997)
6. Chemudugunta, C., Holloway, A., Smyth, P., Steyvers, M.: Modeling documents by combining semantic concepts with unsupervised statistical learning. In: ISWC. (2008) 229–244
7. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI 2010. (2010)
8. Maedche, A.: Emergent semantics for ontologies. In: Emergent Semantics. IEEE Intelligent Systems. IEEE Press (2002) 85–86
9. Aberer, K., et. al: Emergent semantics principles and issues. In: DASFAA 2004, Springer (2004) 25–38
10. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
11. Taubert, J., Hindle, M., Lysenko, A., Weile, J., Köhler, J., Rawlings, C.J.: Linking life sciences data using graph-based mapping. In: DILS, Springer (2009) 16–30
12. ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics (2005) 79–115
13. Doorenbos, R.B.: Production Matching for Large Learning Systems. PhD thesis (1995)
14. Pesquita, C., Faria, D., Falco, A.O., Lord, P., Couto, F.M.: Semantic similarity in biomedical ontologies. PLoS Computational Biololgy **5**(7) (2009)