DEW BEADS: A Framework for Distributional and Emergent Web Semantics*

Vít Nováček[1], Tudor Groza[2], Siegfried Handschuh[1]

[1]Digital Enterprise Research Institute (DERI), National University of Ireland Galway
[2] eResearch Lab, School of ITEE, The University of Queensland, Australia
E-mail: vit.novacek@deri.org

Abstract. This is an extension of an accepted ISWC'11 research track contribution [1], which introduces an alternative, bottom-up and emergent conception of the Web semantics based on the distributional hypothesis (an approach that is rather complementary to the top-down Semantic Web standards based on logics and model theory). The promising potential of our proposal has been demonstrated in [1] by a thoroughly evaluated experiment in knowledge consolidation. In this more technically oriented demo paper, we augment [1] by an overview of DEW BEADS – an open source framework we implemented to test our research ideas[1]. We describe the framework's architecture and features in Section 2. Section 3 then shows an example of DEW BEADS deployment to exploration of knowledge in life science publications. We also outline the general framework's usage in custom applications there.

1 Introduction

The traditional Semantic Web approaches are largely top-down – the meaning of resources (i.e., things) on the Web is being described by various authors and the resulting descriptions are published as RDF data or more expressive ontologies (usually becoming a part of the growing Linked Data cloud). Machines can then use the fruits of this continuous process to work with the Web content more efficiently by exploiting its explicitly captured meaning. Yet, in [1] we argue that the traditional top-down approaches cannot tackle the most substantial part of the Web's meaning, which is not being *asserted* by particular authors, but rather *emerges* from the Web content itself in a distributed manner.

To fill this critical gap in the current Semantic Web research, we have proposed and implemented an alternative, bottom-up approach to representing and exploring the meaning of the Web [1]. The approach stems 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" [3]. In

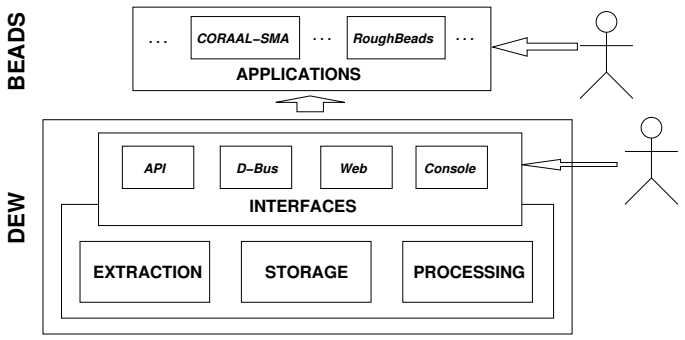
* This work has been supported by the 'Líon II' project funded by SFI under Grant No. SFI/08/CE/I1380.

[1] Note that DEW BEADS is essentially a substantially generalised and revamped successor of our EUREEKA and CORAAL projects described for instance in [2].

the Web context, we can rephrase this to characterise the meaning of a thing 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 [4] and adapted to the Semantic Web specifics. We also show how one can execute rules on the top of the tensor representation, which effectively leads to a smooth combination of the bottom-up (distributional) and top-down (symbolic) approaches to the representation of meaning. The resulting framework can be used for clustering of related entities or properties, semantic search, incremental induction and execution of rules, or discovery of analogies among Semantic Web data. These tasks are rather varied, yet with our approach, one can still tackle them by applying well-founded linear algebra and statistical data analysis methods to different perspectives of the underlying tensor representation [1].

2 DEW BEADS Internals

The core part of the presented framework is DEW (an abbreviation of Distributional, Emergent and Web, key features of our approach to semantics). DEW is a set of Python modules implementing various services for extraction of knowledge in the form of triples, representation and storage of triples augmented by certainty and provenance, and for processing and analysis of the stored content. DEW can be used for building various BEADS – front-end applications catering for Browsing, Exploration, Augmenting, Delivery and Searching of the processed content. The architecture of the framework is illustrated in the figure below. There are four main library modules in DEW. (1) The *extraction* module



implements a simple method for NLP-based extraction of RDF statements from English natural language texts. (2) The *storage* module implements a store for the distributional tensor representations of the Semantic Web data on the top of a relational database (MySQL, PostgreSQL and SQLite back-ends are currently

supported). The module also contains methods for import and export of RDF graphs or N3 rules, and for manipulation of the store data structures (e.g., computation of matrix perspectives from the main tensor, or matrix dimensionality reduction). In addition, a full-text index and ranking for querying the store can be computed by the module. (3) The *processing* module contains algorithms for analysing the store (e.g., identification of similar entities or rule learning) and for rule-based materialisation. (4) The *interface* module exposes DEW to other applications or directly to users. In addition to a comprehensive library API, we provide a D-Bus (c.f. `http://en.wikipedia.org/wiki/D-Bus`) server for applications installed on the same machine. A simple HTTP server exposes querying of the DEW content to external applications via a Web service. Finally, a console interface allows for using the back-end functions via a command interpreter.

On the top of the DEW interfaces, one can deploy various BEADS applications for exploration of the content processed by the back-end. One example is CORAAL-SMA, a prototype engine for intelligent search in publications about spinal muscular atrophy (see the next section). Another example is *RoughBeads*, a readily deployable Python Web interface for semantic search and faceted browsing we are currently developing in order to let users easily explore their data sets within their own DEW BEADS installations.

3 Using DEW BEADS

DEW can be used simply by importing the corresponding Python modules. Development snapshots can be downloaded at `http://140.203.154.177/`. One can also find there a web service exposing querying of a couple of DEW stores we are working with (e.g., the experimental data set from [1]). More stable versions of the sources are soon to be provided at `http://pypi.python.org/pypi/dew` (together with the *RoughBeads* front-end and detailed documentation materials). In the following we give examples of the DEW BEADS practical usage.

Searching and Browsing for SMA Knowledge This demo represents an application of DEW BEADS in the Spinal Muscular Atrophy (SMA) domain. As a part of a joint endeavour by DERI, eResearch group of The University of Queensland, Elsevier, SMA foundation and ISI group at USC, we have recently employed DEW to deliver CORAAL-SMA, a search engine for SMA publications. It allows for semi-automated discovery of related knowledge scattered across isolated scientific articles. DEW is used for extraction of statements from raw text (together with a complementary triple extraction tool developed at ISI), storage, integration and analysis of the extracted content, discovery of implicitly related entities and exposing of the resulting knowledge base to users via an interface for searching and faceted browsing (the UI is a Java application communicating with DEW via the D-Bus interface). See `http://goo.gl/08Jri` for the CORAAL-SMA interface and `http://goo.gl/87Sbi` for documentation.

In a nutshell, CORAAL-SMA allows users to ask full-text queries as in any other search engine, however, the queries are evaluated against the statements automatically extracted and inferred from the SMA texts. The results are pre-

sented in a rich, multi-faceted browsing interface to investigate the interesting discovered relations in detail. Further context of the result statements can be checked directly in CORAAL-SMA in the abstracts of related publications, or by visiting the linked PubMed pages of the articles. As all the results can be browsed along their particular facets (subjects, predicates, objects and/or publication sources), one can quickly focuse on particular details of interest.

General Usage If one wants to use the DEW BEADS framework for analysing their own data, the first step is to download and configure the necessary software (via `http://140.203.154.177/` or `http://pypi.python.org/pypi/dew`). The typical usage pipeline is then as follows: (1) text-to-RDF extraction (optional – one can also merely import extant RDF data for analysis); (2) import of RDF data; (3) computation of a tensor representation of the data (a "statement corpus", knowledge base); (4) corpus materialisation (optional); (5) computation of corpus perspectives (matrices supporting the knowledge base analysis); (6) reduction of the perspectives' dimensionality (optional, but recommended, as it generally reduces noise and speeds up some types of analysis in case of large knowledge bases); (7) knowledge base analysis (such as clustering of similar entities and properties, or learning of rules from the data); (8) export of the analysis results as RDF data and/or their incorporation into the knowledge base; (9) full-text indexing and ranking of the knowledge base; (10) querying and browsing of the knowledge base (in the DEW console interface, in *RoughBeads*, or via a custom BEADS front-end developed on the top of DEW).

4 Conclusion and Future Work

In this short paper, we have outlined DEW BEADS, which is a suite of software tools implementing an alternative, bottom-up approach to Web semantics introduced in [1]. Our conference demonstration will mainly consist of the CORAAL-SMA front-end presentation. Depending on the audience interest, we can also show how to deploy and set up a custom DEW BEADS installation.

In future, we are going to continuously incorporate newly researched emergent knowledge extraction and processing techniques into DEW BEADS. In the short-term horizon, we intend to publish a first stable version of the framework as a readily deployable Python package, so that everyone can easily make use of it within their own Semantic Web data analysis tasks.

References

1. Nováček, V., Handschuh, S., Decker, S.: Getting the meaning right: A complementary distributional layer for the web semantics. In: Proceedings of ISWC'11, Springer (2011) In press. Pre-print available at: `http://goo.gl/FRT77`.
2. Nováček, V., Decker, S.: Towards lightweight and robust large scale emergent knowledge processing. In: ISWC'09, Springer (2009)
3. Firth, J.: A synopsis of linguistic theory 1930-1955. Studies in Ling. Anal. (1957)
4. Baroni, M., Lenci, A.: Distributional memory: A general framework for corpus-based semantics. Computational Linguistics (2010)