Language Technologies and the Evolution of the Semantic Web

Enrico Motta, Marta Sabou

Knowledge Media Institute, The Open University, Milton Keynes, UK
{e.motta, r.m.sabou@open.ac.uk}

Abstract

The availability of huge amounts of semantic markup on the Web promises to enable a quantum leap in the level of support available to Web users for locating, aggregating, sharing, interpreting and customizing information. While we cannot claim that a large scale Semantic Web already exists, a number of applications have been produced, which generate and exploit semantic markup, to provide advanced search and querying functionalities, and to allow the visualization and management of heterogeneous, distributed data. While these tools provide evidence of the feasibility and tremendous potential value of the enterprise, they all suffer from major limitations, to do primarily with the limited degree of scale and heterogeneity of the semantic data they use. Nevertheless, we argue that we are at a key point in the brief history of the Semantic Web and that the very latest demonstrators already give us a glimpse of what future applications will look like. In this paper, we describe the already visible effects of these changes by analyzing the evolution of Semantic Web tools from smart databases towards applications that harness collective intelligence. We also point out that language technology plays an important role in making this evolution sustainable and we highlight the need for improved support, especially in the area of large-scale linguistic resources.

1. The Evolution of the Semantic Web

The key intuition underlying the Semantic Web (Berners-Lee et al., 2001) is that the availability of huge amounts of formally described semantic markup, at a scale comparable to that of the current Web, will make it possible to achieve a dramatic improvement in terms of agent interoperability and user functionalities, which will be enabled by the technology. For instance, the application of semantics to Web Services (Hepp et al, 2005) will make it possible to achieve flexibility at scale, where services will be dynamically located, composed and executed, a process which currently is carried out manually and is therefore expensive. It is also easy to envisage all sorts of new 'smart' functionalities for Web users, which will become possible once semantic markup becomes truly ubiquitous. For instance, we will see new tools for business intelligence, new shopping services, and new forms of news generation, syndication, and personalization, just to list a few examples.

Unfortunately, while this is a compelling and exciting vision, we are still quite a long away from realizing it. In particular, one obvious problem is that all these applications require that a large-scale Semantic Web is there in the first place, which is not the case yet. At the same time it is obvious that the process of realizing the Semantic Web cannot proceed according to a waterfall model, whereby we first build the required infrastructure and produce large-scale semantic markup, and in a second phase we exploit such markup to produce exciting new applications. Clearly, the two processes, Semantic Web construction and application (or at least, demonstrator) development, have to go hand-in-hand. The result of this strategy so far has been that, by and large, the early demonstrators produced in the past few years lack many of the key elements which will characterize 'real' Semantic Web applications. Specifically, Semantic Web applications will operate in an ***open, large-scale, distributed and heterogeneous*** environment, while these early 'proof-of-concept' tools provide semantic techniques on top of rather small, homogeneous and centralised data stores. Consequently, they are more akin to traditional knowledge-based systems, than to 'real' Semantic Web applications.

Recently, however, we have reached a turning point in the history of the Semantic Web regarding its size and development. The Semantic Web is gaining momentum by registering a 300% growth in 2004 alone and thus outpacing the growth of the Web itself (Lee & Goodwin, 2004). There is now a reasonable amount of online semantic data, to such an extent that the need has arisen for a semantic search engine, Swoogle (Ding et al., 2005), which can crawl and index all these data. Hence, we are now slowly reaching a key point in the history of this very young discipline, where we can move away from the early, simplified applications and start developing the kind of applications, which will characterise the Semantic Web of the future. In this paper we will analyse the current state of the art of Semantic Web applications and in particular we will look at a number of existing demonstrators, with the aim of identifying and differentiating the elements typical of first-generation Semantic Web applications, from those which will characterise the 'real' Semantic Web. In the analysis we will emphasize the key role played by language technologies in the context of the Semantic Web and in the second part of the paper we will identify some key 'missing bits', especially in the area of large scale linguistic resources, which need to be more closely targeted to the new scenarios presented by the Semantic Web.

2. From Smart Databases to Harnessing Collective Intelligence

Magpie (Dzbor et al., 2003), was one of the first tools to envision new mechanisms for browsing and making sense of information on the Semantic Web. In the absence of available semantic markup, this tool automatically generates a semantic layer, by mapping items on the current web page to an ontology, by means of Named Entity Recognition technology. In this respect Magpie is a classic example of a first-generation tool. Because Magpie

assumes that the Semantic Web does not yet exists, it generates one on the fly, using appropriate linguistic technology. While this is obviously a clever way of bootstrapping semantic browsing, the limitation is that only one ontology at the time is active and only one semantic layer is generated for a given web page. Having said so, in contrast with other first-generation tools, Magpie is not domain-dependent and indeed it is easy for a user to switch from one ontology to another, while maintaining the constraint that only one ontology is active at the time and this has to be explicitly selected by the user. In contrast with this approach, next-generation Semantic Web browsers should be able to bring in relevant markup from different sources, according to different ontologies, in a dynamic way.

In 2004 the annual Semantic Web Challenge was launched, whose first winner was CS Active Space (Schraefel et al., 2004). This application gathers and combines a wide range of heterogeneous and distributed Computer Science resources to build an interactive portal. The top two ranked entries of the 2005 challenge, Flink (Mika, 2005) and MuseumFinland (Hyvonen et al., 2005), are similar to CS Aktive Space as they combine heterogeneous and distributed resources to derive and visualize social networks and to expose cultural information gathered from several museums respectively. However, there is no semantic heterogeneity and no 'openness' here: these tools simply extract information to populate a single, pre-defined ontology. A partial exception to this rule is Flink, which makes use of some existing semantic data, by aggregating online FOAF files.

Another interesting first-generation Semantic Web tool is AquaLog (Lopez et al., 2005), an ontology based question answering system that interprets a question asked using natural language and uses the structure and instances of an ontology to answer it. Like Magpie, AquaLog is ontology-independent, however, it can only use one ontology at the time.

Obviously, the major challenge faced by these early tools and applications was the lack of online semantic information. Therefore, in order to demonstrate their methods, they had to produce their own semantic metadata, before being able of utilizing them. As a result, either the focus is on a single, well defined domain (Flink, CS AktiveSpace, MuseumFinland), or the tool is domain-independent, but only one ontology can be active at the time (Magpie, AquaLog). Having established a core ontology, data extraction is carried out by defining the appropriate scraping mechanisms, while integration is automated to some extent by relying on some domain specific heuristics. Considerable attention is paid to ensure a high quality of the semantic data, for example, by correctly fusing similar instances.

Taking a step back it is easy to see that all these applications follow closely the paradigm of database centered applications. Although they set out to integrate distributed and heterogeneous resources, these resources sooner or later end up in a centralized semantic repository aligned under a single ontology (playing the role of the database schema). Obviously this is more a constraint imposed by the environment rather than a deliberate choice of the application builders. The use of intelligent reasoning techniques to harvest the collected semantic data promotes these early applications to the status of smart databases.

Luckily, as already pointed out, the Semantic Web is gaining momentum and recently we have seen the emergence of new tools, which already instantiate some of the features which will characterize the next generation of Semantic Web tools. Here we will focus on two of these 'new generation tools', PiggyBank (Huynh et al., 2005) and PowerAqua (Lopez et al., 2006). PiggyBank allows users to collect semantic information while browsing the Web and then analyze and share this information within a community. PowerAqua moves away from the limitations of AquaLog and provides question answering support on an open, distributed and heterogeneous Semantic Web. In what follows, we will use these two tools to illustrate what we regard as the key features of the next generation of Semantic Web tools.

Decoupling the process of engineering from that of exploiting the Semantic Web. While previous tools had to engineer the semantic data before utilizing them, both PiggyBank and PowerAqua assume that they operate in an environment characterized by large scale, distributed semantic markup. Nevertheless PiggyBank still provides some scraping functionalities that can be invoked by users to acquire semantic information when this is not available.

Operating with heterogeneous semantic markup and multiple ontologies. Both PiggyBank and PowerAqua drop the single-ontology assumption and assume they have to deal with heterogeneous semantic markup. However, there is an important difference here between PowerAqua and PiggyBank: the former provides methods to integrate such heterogeneous information, while the latter does not.

Openness with respect to semantic resources. In the case of first-generation tools, it was very difficult to add new sources or integrate new ontologies. This is not an issue for either PowerAqua or PiggyBank. PowerAqua simply does not include any limiting assumption on the data that can be available to answer a query. PiggyBank allows the user to select the data that she is interested in during browsing. While in the case of other tools adding a new data source involved serious adaptation of the system, PiggyBank allows this simply with a mouse-click.

Scale more important than quality. We have seen that a lot of the emphasis in first-generation tools was on quality: given an ontology and a set of extraction methods, the goal was typically to populate the ontology in the most quality-controlled way. This was possible because of the relatively 'closed' nature of these applications. However, when operating at scale on a large and distributed Semantic Web, then it becomes much more difficult, if not impossible, to ensure strict quality controls. As a result both PiggyBank and PowerAqua take a lightweight approach to ensuring data quality. For instance, PiggyBank does not attempt to merge similar instances. In contrast with PiggyBank, PowerAqua does need to deal with the co-reference problem in order to provide meaningful answers, however, it is agnostic to the quality of the available semantic information. It just tries to find relevant semantic markup that can be used to answer queries. As such it moves away from traditional quality-centered expert systems, just as the Web differentiated itself from hypertext, by allowing broken links. In our view this is no accidental phenomenon, but an indication that, as in the case of the Web, the strength of the Semantic Web will be more a by-product of its size than its absolute quality.

WWW – We Want Web! Just like the first cars ever produced were far more similar to the old horse-driven carts than to today's sleek cars, we have pointed out that early Semantic Web applications are far more similar to the classic knowledge-based systems, than to the Semantic Web applications of the future. However, we are now witnessing Semantic Web applications which move away from first generation tools and try to bring the Semantic Web closer to the Web. In particular PiggyBank follows the current trend of community driven portals and allows users to tag their resources and then share these tags with their community to foster collaborative work and resource discovery. PiggyBank even goes one step further than traditional tag collections (folksonomies) by representing tags as RDF resources rather than text snippets. This opens the possibility of more complex operations with tags, such as establishing relations between them. CONFOTO (Nowack, 2005) is another application, which embraces the tagging based community portal view.

Existing applications already integrate Web Services in their functionalities: for instance, Flink uses Google and Google Scholar as services, while Magpie allows the invocation of services that are available for the entities that it discovers on a web page. Indeed, the Web itself is increasingly populated with services that provide a large variety of functionalities to users. PiggyBank can acquire semantic data either by copying existing data or by running some scrapers on Web pages. These scrapers are in essence basic services. However, rather than using a set of previously selected services (such as Flink does), PiggyBank can work with any scraping services provided by sites or by third party users.

From intelligent applications to harvesting collective intelligence. We have already highlighted several differences between traditional knowledge-based systems and 'real' Semantic Web applications, to do with quality issues, the degree of control over the domain data, the use of single vs. multiple ontologies, etc. There is also another crucial difference, which relates to alternative notion of machine intelligence. In traditional knowledge-based systems intelligence is normally associated with the reasoning ability of a system, e.g., its ability to carry out diagnosis, or planning, or scheduling or some other complex task. The view of 'intelligence' embodied by PowerAqua and PiggyBank is different. These systems of course embed smart Artificial Intelligence technology to extract data from web sources (PiggyBank) and to identify and integrate the data needed to answer a complex query (PowerAqua). However, here intelligence is also a by-product of operating with large amounts of data. The users act as catalysts in deriving value from collectively gathered, tagged and shared semantic data, thus using the system to harvest collective intelligence.

3. Where do we want to go?

Based on the considerations above, we can conclude that the embryonic emergence of a Semantic Web has already caused a paradigm change in the way applications and tools are developed. But what lessons can we learn from this analysis? Can they help to extrapolate what will be important in the future Semantic Web?

Tim O'Reilly derives the following success criterion that differentiates Web2.0 software: *"the value of the software is proportional to the scale and dynamism of the data it helps to manage"* (O'Reilly, 2005). This principle is consistent with our view that scale rather than pure quality is likely to be a key success factor for the future Semantic Web. As a side effect of the growth of the Semantic Web, there will also be a continuous tendency to move towards applications that utilize existing semantic data rather than having to generate their own. Because these data will be heterogeneous, the complexity of the tools will be a function of their ability to make sense of such heterogeneity.

In addition to the technological issues, the growth of the Semantic Web will also introduce new cultural, social and political issues. For instance Buitelaar et al. (2003) point out that current Semantic Web tools do not take sufficiently into account the multicultural and multilingual nature of Web data; Motta (2006) discusses the dangers that "dominant conceptualizations" might pose to the democratic nature of knowledge publishing on the web; and finally O'Hara (2004) details the socio-technological problems which might arise as a side-effect of the limits of formal knowledge representation.

4. Language Technologies for Engineering and Using the Semantic Web

Language technologies and the Semantic Web can mutually benefit from each other (Buitelaar et al., 2003). Following up from the above analysis on the status of the Semantic Web we highlight what we regard as some 'missing pieces', which are needed by the semantic web enterprise. Needless to say, we do not claim that our analysis is comprehensive. Indeed, such an attempt would require much more space than is feasible here and in any case it is simply not possible to envisage all the possible ways in which language technologies can be harnessed to support Semantic Web research and development. Here we only wish to propose some concrete examples in which these technologies play an important role and highlight the gap between what is available and what is needed.

The importance of using and combining several language resources and processing algorithms has already been recognized while building the first generation Semantic Web tools. For example, Magpie uses a Named Entity Recognizer to automatically generate semantic markup, while AquaLog uses GATE (Cunningham et al., 2002) and WordNet (FellBaum, 1998) to translate queries into a logical form and to try and map the user terminology to that used by the current ontology. One of the lessons that we learned from these applications was that existing language resources are limited in size and heterogeneity when used in semantic web tasks. Even WordNet, the largest and most widely used language resource, has limited topic coverage. Furthermore, WordNet proved insufficient when used to disambiguate the sense of relations, and not just concepts, a task often performed by AquaLog. Indeed, when dealing with relational data, language resources focused on verbs, such as VerbNet (Kipper et al., 2000) or FrameNet (Baker et al., 1998), are in principle more suitable. However their breadth is even more limited than that of WordNet.

The problem of *knowledge sparseness* (Sabou, 2006) is not only evident in traditional language resources but it also characterizes the current online semantic data collection. Our preliminary experiments with Swoogle, show an uneven distribution of knowledge over topic

domains: some domains (e.g., academia, bioinformatics) are well covered while other domains are not even mentioned. Indeed, even if the Semantic Web is gaining momentum, its scope is still rather limited (Sabou, 2006).

One way to overcome this knowledge sparseness problem would be the semantic enrichment of resources such as thesauri and folksonomies. Some Semantic Web tools already rely on the enrichment of such primarily textual sources. For example, the developers of MuseumFinland carried out a significant manual effort to enrich, merge and populate several thesauri (Hyvonen, 2005).

A particularly interesting case is that of *folksonomies*, collectively engineered tag sets used as a mechanism to facilitate sharing and searching for information in online community portals. Folksonomies have been widely adopted because the tagging mechanism, even if semantically inferior to ontologies, allows an intuitive browsing of the information collection. However, such a tag-based search is analogous to keyword search and therefore limited. Nevertheless, there is obvious potential in integrating the advantages in terms of rapid annotation support provided by folksonomies with the formal semantics and rich structures provided by ontologies. On the one hand, folksonomies could be disambiguated and enriched with formal knowledge, by using the large pool of ontological knowledge that has already been created. On the other hand, the social dynamics of a collaborative folksonomy development process could be a key element for extending and evolving knowledge captured in ontologies or other language resources.

Language technology could also be used to support the semantic annotation of Web Services. For example, natural language processing and classification techniques were used by at least two different researchers (Hess, 2005; Sabou, 2005) to automate the task of Web service annotation. Unfortunately, this research was hampered by the lack of language resources (e.g., corpora, verb-centered lexicons) and evaluation methods suitable for the context of Web Services. The corpora that were produced as a side effect of the actual research are different from traditional corpora as they are small collections of (very) short documents written in a technical sub-language. As a general lesson, we foresee that applying Semantic Web research in new domains will need (and eventually produce) novel types of language resources, e.g., collections for Short Text Messages (SMS) in mobiles.

5. Conclusions

The Semantic Web is rapidly becoming a reality. However, there is still a great deal that needs to be done. In particular the Semantic Web urgently needs improved large-scale language resources, which can help to address the knowledge sparseness problem and facilitate the integration of heterogeneous data. We believe that not only the Semantic Web is an exciting, compelling and potentially ground-breaking enterprise, but it also provides an important context in which to develop and apply novel computational linguistics technology.

6. References

Berners-Lee, T., Hendler, J., Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34 – 43.

Baker, C., Fillmore, C., Lowe, J. (1998) The Berkeley FrameNet project. In *Proceedings of the COLING-ACL*.

Buitelaar, P., Declerck, T., Calzolari, N., Lenci, A. (2003) Language Resources and the Semantic Web. In *Proceedings of the ELSNET/ENABLER Workshop.*

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the ACL*.

Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P. (2005) Finding and Ranking Knowledge on the Semantic Web. In *Proceedings of ISWC*, p. 156 – 170.

Dzbor, M., Domingue, J., Motta, E.(2003) Magpie - towards a Semantic Web browser. *Proceedings of ISWC*.

Fellbaum, C. (Ed.). *WordNet: an ElectronicLexical Database*. MIT Press, 1998.

Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D. (2005) Semantic Business Process Management: Using Semantic Web Services for Business Process Management. In Proceedings of the *IEEE ICEBE*.

Hess, A. (2006) Supervised and Unsupervised Ensemble Learning for the Semantic Web. *PhD Thesis.*

Huynh, D., Mazzocchi, S., Karger, D. (2005) Piggy Bank: Experience the Semantic Web Inside Your Web Browser. In *Proceedings of ISWC*.

Hyvonen, E., Makela, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S. (2005). MuseumFinland – Finnish Museums on the SemanticWeb. *Journal of Web Semantics*, 3(2).

Kipper, K., Dang, H.T., Palmer, M. (2000) Class-based construction of a verb lexicon. In Proceedings of *AAAI.*

Lee, J., Goodwin, R. (2004) The Semantic Webscape: a View of the Semantic Web. *IBM Research Report*.

Lopez, V., Motta, E., Uren, V. (2006) PowerAqua: Fishing the Semantic Web. In *Proceedings of ESWC*.

Lopez, V., Pasin, M., Motta, E. (2005) AquaLog: An Ontology-portable Question Answering System for the Semantic Web. In *Proceedings of ESWC*.

Mika, P. (2005) Flink: SemanticWeb Technology for the Extraction and Analysis of Social Networks. *Journal of Web Semantics*, 3(2).

Motta, E. (2006) Knowledge Publishing on the Semantic Web: An Optimistic Socio-Technological Analysis. *IEEE Intelligent Systems*. To appear.

Navigli, R. (2005) Semantic Enrichment of Large-Scale Linguistic Resources, *In Proceedings of MEANING-05*.

Nowack, B. (2005) CONFOTO: A Semantic Browser and Annotation Service for Conference Photos. In *Proceedings of ISWC*.

O'Hara, K. (2004) Ontologies and Technologies: Knowledge Representation or Misrepresentation, In SIGIR Forum, 38(2).

O'Reilly, T. (2005) What Is Web 2.0. *http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/ what-is-web-20.html*

Sabou, M. (2005) Building Web Service Ontologies. *PhD Thesis, Vrije Universiteit Amsterdam*.

Sabou, M., Lopez, V., Uren, V., Motta, E. (2006) Ontology Selection on the Real Semantic Web: How to Cover the Queen's Birthday Dinner? *Submitted.*

Schraefel, M.C., Shadbolt, N.R., Gibbins, N., Glaser, H., Harris, S. (2004) CS AKTive Space: Representing Computer Science in the Semantic Web. In *Proceedings of the 13th International World Wide Web Conference.*