

# Integrating Knowledge for the Semantic Web via a weak evidential approach.

Sam Chapman, Fabio Ciravegna,

Department of Computer Science, University of Sheffield  
Regent Court, 211 Portobello Street, Sheffield S1 4DP, UK  
{ S.Chapman, F.Ciravegna } @dcs.shef.ac.uk

## 1 Introduction

The Semantic Web, SW, requires semantically-based structured content both to enable better knowledge retrieval and empower semantically-aware agents. One prerequisite for the SW is the widespread adoption of structured knowledge.

Existing web publishers will not be eager to provide structured content without seeing any added benefits of doing so. This is a classic chicken and egg problem. So without any universal acceptance, other automated methods need to be employed to provide structured content from existing unstructured content. Current technologies available for creating structured content are typically based on human-centred annotation, very often completely manual, of documents.

Manual annotation is monotonous, time-consuming and can introduce noise [1], being incomplete or incorrect, hence decreasing the quality of the information. For these reasons, (and those of scaling up), we believe that the SW needs automatic methods for annotating content. Automatic annotation services such as SemTag[4] and Armadillo[5] intend to automatically provide SW content. One problem with automatically providing content is that no matter how good a system, errors will still be slowly introduced. The old Armadillo version reduced errors by simply integrating the information found from numerous sources choosing the most prevalent instances as the most likely[5]. This approach, although useful, is prone to spamming and problems with common misconceptions. In this paper we introduce the new Armadillo approach, *evidence building*, that adds a degree of trust from the similarity and provenance of instances.

The Armadillo approach of *evidence building* works through a combination of individually weak evidential tests to validate found instances across data sources. The armadillo tests are mostly simple Similarity Metrics, (these are detailed in section 3). The paper then focuses on presenting the Armadillo tool and details a simple use case relating the methodologies used in order to Integrate elicited knowledge more effectively.

## 2 Integration as the Issue

The combination of information instances is far from a new problem, it has been researched in many differing fields for many years. The techniques of In-

Information Integration have been given a number of names, Information Integration, Record Linkage, Deduplication, Object Consolidation, Information Fusion, Merge/Purge, Deduplication, Data Cleaning, Referential Integrity and many more. This is still however a far from solved subject, the Semantic Web throws new challenges and even greater opportunities to resolve this problem, the graph structure and provenance of data allows greater abilities to use external resources for Integration requiring the problem to be readdressed for the Semantic Web. This paper sets out a framework for this in the Armadillo system.

### 3 SimMetrics: Similarity Library

SimMetrics is an open source extensible java library developed for use in Armadillo which contains numerous Similarity Metrics<sup>1</sup>.

A Similarity Metric is an algorithm that, given two inputs, typically strings, returns a numeric measure of their similarity. Similarity measures come from a variety of disciplines, including statistics, DNA analysis, artificial intelligence, information retrieval, information integration and the database community. They are usually simple algorithms e.g. Needleman-Wunch Distance[9], Smith-Waterman-Gotoh Similarity[6], Monge-Elkan Similarity[8] and so on (a full list and descriptions of each similarity method is beyond the scope of this paper and should be sought elsewhere<sup>2</sup>).

Similarity Metrics within the SimMetric library can take as input two textual strings and return a similarity measure normalised to a float ranging between 0.0 and 1.0, 0.0 being entirely different, 1.0 being identical.

String Metrics have been used before in combining data,[3]; but never, (at the time of writing to the readers knowledge), in an extensible architecture for the specific purposes of Integration for the Semantic Web, i.e. taking into account the provenance and linking of Semantic data. The Armadillo tool itself is now detailed, before a specific use case.

## 4 Armadillo

Armadillo is a tool developed within the University of Sheffield; the initial version was proposed by Alexiei Dingli[5], all subsequent versions have been developed by the paper's author.

### 4.1 What is Armadillo?

Information can be available in different formats on the Web: in documents, in repositories (e.g. databases or digital libraries), from agents, web services, web based API's, etc. Information can be extracted from different sources with

<sup>1</sup> <http://sourceforge.net/projects/simmetrics/>

<sup>2</sup> <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>

differing reliability; information found can even be spam, typos, deliberate misinformation or simply erroneous. When information is contained in textual documents, extracting it requires more sophisticated IE methodologies based on linguistic analysis and methods to ensure a degree of reliability of extracted information.

Information from whatever source is often redundant, i.e. that it can be found in different contexts and in different superficial formats - the redundancy of information can in itself be a weak proof of its validity [5]. However, we suggest, that this alone is not enough, additional information either within or surrounding extracted entities can be used to develop additional evidence. Armadillo now uses an evidence building approach of numerous rudimentary techniques, (based on SimMetrics, Section 3).

## 4.2 How Armadillo works?

Armadillo learns how to best extract information in the following way:

1. it mines a coherent portion of the repository (e.g. a web site or a class of sites);
2. it integrates information and assigns reliabilities of different sources (e.g. digital libraries, services, webpages). These ratings are used to *direct* the learning from the repository;
3. it discovers new information in the repository that in turn is rated and used to bootstrap new learning until a stable information base is reached;
4. it stores the harvested information into a RDF Knowledge base. The database can then be used to access the extracted information (as detailed later) or to produce indices for document retrieval or annotation.

Armadillo is a data driven system typically beginning from rigidly structured reliable sources using examples provided by either a wrapper, the user or previous data. Armadillo uses previously obtained seeds to learn on more complex sources (e.g. free texts) using the previously acquired information.

In order to explain in more depth the working of Armadillo an example is now detailed.

## 4.3 The Computer Science Department Application

Consider the following example task of mining websites of Computer Science Departments to find academics (name, position, home page, email address, telephone number and a list of publications more complete than the one provided by repositories such as Citeseer).

Simply discovering who works for a department is more complex than generic Named Entity Recognition (NER) as many irrelevant people's names are mentioned in a site, e.g. names of undergraduate students, secretaries, as well as names of researchers from external sites and hence irrelevant for this task.

Armadillo uses a statistical evidence based looping approach to aid the validation task. Initially a quick list of potential names of people working in the

department is found - this can be generated from a gazetteer, manual annotation, a wrapper or via simple crawling and NER. The collection of initial potential data need not be considered a gold standard (although a reasonable reliability is required) as more evidence is then sought for each potential academic.

Armadillo then loops through two repeating phases:

1. Evidence Building and Validation - relies upon building up a series of weak evidences to cumulatively rate knowledge for accuracy.
2. Extraction of potential knowledge - uses ML to learn from validated existing knowledge to find contextual patterns in rated data sources in order to elicit more potential knowledge which must then be validated.

**Evidence Building and Validation** - comes from the series of weak approaches each of which are combined to give a rating which can be used to validate knowledge. Firstly additional redundant sources of the same data are identified via a simple search, e.g. google finding relevant URLs. This redundancy by itself is no longer considered valid evidence as the reliability of sources themselves must also be taken into account, e.g. a source with lots of potential academics is considered a better source than a source detailing just a single academic.

The rating of sources influences Armadillo's perceived validity of the potential academics found. Using the previously mentioned SimMetric library, section 3, the extracted entities are cross examined for simple similarities: if something is suitably dissimilar from the rest, for example a failed capture 40 times longer than normal, it is considered more possibly an error and its validity rating is decreased as well as the rating of the source from which it was found. At present the cross similarity tests are a combination of SimMetric's similarity metrics, although more complex approaches could be employed. The context around a capture can also detail likelihood, e.g. the similarity of Part of Speech tags surrounding instances, various other similarity techniques to provide further weak evidence could be used, for example a vector space model of the source (e.g. webpage) similarity, a comparison of similarities in a link analysis of data sources or an analysis of the source documents DOM structure and its similarity to other sources.

This combination of multiple weak techniques can provide improved confidence in the extracted knowledge Armadillo finds, (again although not yet implemented it is envisioned to prioritise techniques through Information Gain and memory or time costs).

**Extraction of potential knowledge** - armadillo as in previous papers[5] integrates the Amilcare [2] (LP)<sup>2</sup> algorithm to extract potential new knowledge, but can be extended to encompass different ML algorithms, for example T-Rex [7]. Using ratings from the evidence building approach (Section 4.3) the Information Extraction learns its contextual rules on the most highly rated sources using the most likely instances as seed data.

This allows ML to use the information of existing finds to suggest more potential entities which are investigated further using *evidence* to rate the found extractions, thus improving precision.

**Trusted sources** - using Armadillo's looping methods, detailed in section 4.2, it quickly becomes apparent that the better structured or more reliable information sources, for example html lists on a regular site and external sources such as citeseer<sup>3</sup> and unitrier<sup>4</sup> are identified as *oracles* to quickly test new entities. This generic evidence based approach can of course be extended to any domain where redundant evidence can be found.

## 5 Conclusion and Future Work

This paper details armadillo's extensible technique for building evidence for Information Integration for the Semantic Web using simple similarity measures. This method is actively used in Armadillo to clean, normalise and even disambiguate data gathered from various sources via ML methodologies.

Future work will add additional metrics and methodologies to the Integration process also investigating learning techniques to prioritise the methods applied to those best suited to a domain.

## Acknowledgements

This work was carried out within the AKT project (<http://www.aktors.org>), sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/N15764/01), and the Dot.Kom project, sponsored by the EU IST asp part of Framework V (grant IST-2001-34038).

## References

1. F. Ciravegna, A. A. Dingli, D. Petrelli, and Y. Wilks. User-system cooperation in document annotation based on information extraction. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*. Springer Verlag, 2002.
2. F. Ciravegna and Y. Wilks. Designing adaptive information extraction for the semantic web in amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2003.
3. W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string metrics for matching names and records. In *KDD Workshop on Data Cleaning and Object Consolidation*, 2003.

<sup>3</sup> <http://citeseer.ist.psu.edu/>

<sup>4</sup> <http://www.informatik.uni-trier.de/~ley/db/>

4. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the World Wide Web Conference 2003*, 2003.
5. A. Dingli, F. Ciravegna, and Y. Wilks. Automatic semantic annotation using unsupervised information extraction and integration. In *Proc. of the K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation*, 2003.
6. O. Gotoh. An improved algorithm for matching biological sequences. In *Journal of Molecular Biology*, volume 162, pages 705–708, 1981.
7. J. Iria. T-rex: A flexible relation extraction framework. In *Proceedings of the 8th Annual Colloquium for the UK Special Interest Group for Computational Linguistics (CLUK'05)*, 2005.
8. A. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *Second International Conference on Knowledge Discovery and Data Mining, (KDD)*, 1996.
9. S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. In *Journal of Molecular Biology*, volume 48(3), pages 443–453, 1970.