Benchmarking ontology-based annotation tools for the Semantic Web

Diana Maynard
Department of Computer Science
University of Sheffield, UK
diana@dcs.shef.ac.uk

Abstract

This paper discusses and explores the main issues for evaluating ontology-based annotation tools, a key component in text mining applications for the Semantic Web. Semantic annotation and ontology-based information extraction technologies form the cornerstone of such applications. There has been a great deal of work in the last decade on evaluating traditional information extraction (IE) systems in terms of performance, but such methods are not sufficient for ontology-based systems. Furthermore, usability aspects need to be considered such as scalability, accessibility and interoperability. We outline the main requirements in terms of both usability and performance, proposing possible solutions and using examples of existing tools and resources.

1 Introduction

In the field of bioinformatics, there has been increasing interest in the use of ontologies, because they provide a means of accessing the information stored in large databases not only by humans (as traditionally was the case) but also by computers. The Gene Ontology (GO)[1] is one of the largest and most important ontologies in the field. By storing terms and their relations and thereby providing a standard vocabulary across many different resources, it enables annotation and querying of databases such as SWISS-PROT. For example, Lord et al. [12] present methods for measuring semantic similarity in GO in order to enable querying of such databases for e.g. proteins semantically similar to a query protein.

In the last decade, methods for information extraction have become extremely important in the field of biomedicine. Detecting gene and protein names in texts is essential for knowledge discovery, and the lack of standardisation and presence of both ambiguous terms and term variation makes the task very complex. Similar mechanisms to carry out such tasks have been used as for traditional open-domain information extraction (i.e. both rule-based and machine learning approaches); however, the increasing use of ontologies has paved the way for the application of ontology-based information extraction techniques in this domain. Development of such applications is hampered by the lack of standardisation and suitable metrics for testing and evaluation.

Until now, ontologies in biology were considered as mere guides for data structure, with the main purpose being to access the most useful documents and articles according to the researcher's interests. Applications such as semantic annotation enable us to combine and associate existing ontologies in the biological field, and to perform an integral modelling of the disparate biological data sources. Once this is achieved, knowledge can be extracted from the data repositories by means of agents, annotations and the semantic grid.

This paper discusses and explores the main issues for evaluating ontology-based annotation tools, a key component in text mining applications for the Semantic Web. Text mining concerns the discovery of facts and structured information from large collections of unstructured documents such as is found on the Web. Semantic annotation and ontology-based information extraction technologies form the cornerstone of text mining applications for the Semantic Web. There has been a great deal of work in the last decade on evaluating traditional information extraction (IE) systems in terms of performance, but such methods are not sufficient for ontology-based systems, as we shall discuss. Furthermore, there are many other aspects of such tools which also need to be evaluated when comparing or establishing the

[1] http://www.geneontology.org

usefulness of different systems, which go beyond the typical requirements for a traditional IE system, in terms of criteria such as usability, scalability, accessibility and interoperability.

In this paper, we shall outline the main requirements for such systems, both in terms of usability, and in terms of performance, proposing possible solutions and using various examples of existing tools and resources.

2 Requirements of ontology-based annotation tools

In this section, we detail some of the main requirements for annotation tools in terms of benchmarking.

2.1 Expected functionality

First, we briefly explain the expected functionality of ontology-based annotation tools, in terms of the minimum that they should be expected to achieve. Annotation tools may fall into several types: manual, semi-automatic or automatic. All three types follow the same basic principles and can be evaluated in a similar way, though various considerations need to be taken into account according to the type. For example, there would be no point in evaluating the accuracy of manual annotation tools, given that we are comparing against a human-produced gold standard. The relative speed of annotation for different systems (both manual and automatic) is, on the other hand, an important criterion. The objective of the tool is that given a corpus of text and an existing ontology, it should be able to create semantic metadata by populating the texts with instances from the ontology. In some cases they may also modify the existing ontology structure or content, for example by adding new concepts or reassigning instances, but this is not obligatory. Semi- and fully automatic annotation tools tend to rely heavily on IE techniques.

Traditional IE is not, however, completely suitable for metadata creation, because semantic tags need to be mapped to instances of concepts, attributes or relations, and this is not always a straightforward process. Also most IE systems based on Machine Learning methods, e.g Amilcare [4], do not deal well with relations, although they are very good at finding entities (which can be mapped to instances in the ontology). On the other hand, there have been few knowledge engineering approaches (which use rule-based systems, e.g. GATE [5]) that deal successfully with relations except in very restricted domains, and these require a substantial amount of time and effort to develop. Rule-based methods have, however, been successfully used for automatic semantic annotation, e.g. in h-TechSight [8] and KIM[11].

2.2 Interoperability

One very important requirement of tools (in most cases) is that they should be interoperable, i.e. that they can be combined with other tools, systems, and datasets and used by different people with different requirements. In particular, the format of the results produced should be usable in other systems and applications, because the results of semantic annotation are not generally useful as a final product, but only when combined with other tools or systems such as information retrieval and other more specialised search facilities, question answering, data evaluation, technology watch and market monitoring, and so on.

Metadata is created through semantic tagging, and can be represented as inline or standoff annotation. Inline annotation means that the original document is augmented with metadata information, i.e. the text is actually modified. Standoff annotation, on the other hand, means that the metadata is stored separately from the original document, with some kind of pointers to the location of the corresponding text in the document. This can be either in the form of a database or as e.g. an XML file. For ontology creation or enhancement, standoff annotation method is generally much better, because the text itself is unimportant, rather it is the information gleaned from the text that is interesting.

Both methods are acceptable from the point of view of interoperability; however, standoff annotation is generally preferable, for the reasons mentioned above, as long as a standard form is used, such as TIPSTER format, or provided that a means of export to a recognised format is provided. This is the problem with inline annotation, because it is difficult to manipulate the annotations once created.

Secondly, the format not only of the data output, but also that of the ontology is important. The tool should ideally be able to use ontologies in different formats such as RDF, OWL, DAML+OIL etc. or at least to provide converters to and from different ontology formats.

Other important aspects of interoperability concern the platform and browser on which the tool runs (e.g. if it only runs on Windows and/or In-

ternet Explorer, whether it is a standalone system, etc.), whether it performs differently when using different browsers, or if there are conflicts when running on certain platforms or browsers. A more detailed study of interoperability amongst annotation systems can be found in [14].

2.3 Usability

Usability is a criterion which is not generally awarded much importance in the research community, but which bears far more significance when evaluating tools for use in industry. Usability includes criteria such as ease of installation and quality of installation documentation, general quality, format and ease of access of documentation for running the software, ease of setup, general aesthetics, and simplicity/complexity of the tasks possible.

Installation should ideally be quick and simple, and documentation for installation should be readily available. It should be clearly separated into sections for different operating systems / platforms if this is applicable.

The main documentation should also be easy to find, not just on installation, but during use of the system. Ideally there should be a method of accessing the documentation directly from the system or tool. Documentation should be available in a non-proprietary format and accessible to all. HTML is usually recommended, PDF is acceptable. Good documentation should include an overview of the system's capabilities, should be well laid out such that information is easy to find, and ideally should include step-by-step descriptions of how to complete tasks (where appropriate) and diagrams and screenshots as necessary.

Setup of the tool is another often overlooked factor in its usability. This is concerned with how easy it is to configure the system to the user's requirements, for example defining paths to ontologies or other resources, changing the fonts and colours appropriately, configuring options such as saving the latest session, altering the look and feel of the tool, e.g. rearranging sidebars, perhaps even changing the default language and/or encoding if appropriate. A usable system will have a wide range of configuration options available so that users can customise the system to their needs. However, these configuration options should be easy to find and clear in their functionality.

Aesthetics is another factor often awarded little importance in research communities. While it may not be crucial to the performance of the system in terms of speed or accuracy of result, it is often a critical selling point in industry. This is linked with the previous point about setup – aesthetics is a subjective matter and therefore the ability to customise the system to one's own requirements or preferences is crucial. Also what may be a matter of taste to one person (e.g. the colour scheme chosen) may be a matter of performance to another (for example a person with sight difficulties may struggle with certain colour schemes).

Finally, the usability of a tool depends on the tradeoff between simplicity and complexity. Ideally a system should be able to perform a wide range of tasks of varying complexities, without sacrificing ease of use. But clearly the more (and more complex) tasks the system can achieve, the harder it is to use. So some kind of middle ground needs to be found.

2.4 Accessibility

Software accessibility is essentially about making tools that are usable, easy to use and attractive to use for everyone (not just for people with disabilities). Generally, however, designing websites and software with certain disabilities in mind covers the majority of cases for people with and without disabilities. Particular care should be taken to design sites and systems usable by the following categories: blind and colour-blind people, people with other visual impairments (e.g. partially sighted); deaf people, people with motor problems (e.g. those who cannot use a mouse or normal keyboard), dyslexic people, people with learning difficulties, people with epilepsy (who may not be able to tolerate flashing colours, for example).

Obviously not all categories need to be considered, depending on the tool and intended user, but care should be taken not to exclude potential unknown users. For example, one might not imagine that a blind person would want to use an annotation tool, but one cannot be sure of this in advance. It is also important not to stereotype certain categories of disability. For example, making sure that tools work with a screen reader will not necessarily benefit all blind and partially sighted people – they may also require easy navigation, clear and simple layouts without clutter, consistency, good use of colour, changeable font sizes, etc.

Some of the most important examples of accessibility problems stem from *inflexibility*. A well designed tool will have options to change the user's preferences regarding colours, layout, font sizes and styles, and so on, and the ability to save and restore latest sessions, etc.

Even though a user should be able to choose such options, the default options should also be well designed. For example, text should be in a mixture of upper and lower case where possible (as this is the most easily readable, hence the reason it is used for road signs), and colour schemes should incorporate dark writing on a light background or vice versa. Icons should be clearly understandable, not just with alternative text on mouseover, but should also use clear symbols and be large enough to click on easily (for those with motor or sight problems). Mouse alternatives should also be widely available, again for people with motor and sight problems, RSI etc.).

2.5 Scalability

Semantic metadata creation can be manual, semi-automatic, or fully automatic. Manual creation is slow and time-consuming, and is therefore unsuitable for large-scale annotation. Semi-automatic methods save time and money, and, like manual methods, are very reliable, but they suffer from the drawback that they represent only one view (that of the user, with respect to a single ontology). They also still require human intervention at some stage in the process (either to train the system by providing initial manual annotation before the system takes over, and/or to verify and correct the results produced by the system). There is almost always a tradeoff between the level of automation, the size of the corpus, and the quality of the final output. Systems which perform well and on large documents are unlikely to be fully automatic; systems which are fully automatic may be able to handle large documents but with lower performance.

Other scalability issues concern storage and manipulation of large ontologies and knowledge bases, and processing speed when dealing with large volumes of data. These issues are specifically addressed in the SWAN project[2], which deals with the problem of massive semantic annotation.

2.6 Reusability

Ideally, annotation systems should be reusable in a wide variety of contexts, i.e. they should work on different kinds of domains and genres. Semi-automatic systems which rely on some degree of manual annotation and/or training can usually be adapted to new domains and ontologies, but will need retraining by the user. This means that they are generally best suited to annotating large volumes of data within a single domain, and in situations where the user has an interest in investing some initial time and effort in the application. They are less suitable for the casual user who wants a ready-made tool to provide instant annotations for his data. Automatic methods, on the other hand, can represent many different views, and they change according to the ontology in qqustion. The IE engine can be retrained for each ontology, and, furthermore, if the ontology changes, they remain up-to-date because the metadata can be regenerated dynamically. However, the tradeoff is that their performance is generally much lower.

Reusability is also linked with interoperability - the more interoperable the tool is, the more reusable it tends to be, because some or all of its components can easily be integrated into other systems.

3 Performance evaluation of ontology-based annotation tools

The benchmarking of ontology-based annotation tools needs to comprise some metrics for evaluation of the quality of output. Such metrics must provide a simple mechanism for comparing different systems and different versions of the same system in a consistent and repeatable way. Evaluation of semi-automatic or automatic annotation tools can be performed by measuring the correctness of the semantic metadata they produce, with respect to a manually annotated set of data (documents) and an ontology, i.e. by evaluating the quality of the information extraction.

The evaluation task for ontology-based information extraction aims to discover in the text all mentions of instances related to the ontology. The gold standard is a set of texts where instances are annotated with their related ontological concepts. We aim to measure how good the IE system is at discovering all the mentions of these instances, and whether the correct class has been assigned to each mention.

3.1 Criteria for defining performance evaluation

When preparing the corpus and metrics, the following are essential [15]:

- Have well defined, written annotation guide-

[2] http://deri.ie/projects/swan/

lines, so that the annotation of the gold standard text is consistent.

- Carry out an analysis of the corpora with respect to distributions of the different tags, and also analysis of the complexity of the domain for the IE task, and a statistical profile of the tasks (i.e., how difficult the task is for the baseline system).
- Ensure that at least some portion of the corpus, if not all of it, is double-annotated, or better still, triple-annotated, and that there is a mechanism for conflict resolution where annotators do not agree.
- Measure inter-annotator agreement (IAA) and publish this so systems can know when theyve reached the ceiling (if people cannot achieve 100% correctness, then it is unlikely that systems ever can).
- Provide a pre-defined split of the corpus into training and testing data. allowing for measuring the statistical significance of the results.

When defining the evaluation metric itself, the following criteria are suggested by [7]. The metrics should:

- reach its highest value for perfect quality;
- reach its lowest value for worst possible quality;
- be monotonic;
- be clear and intuitive;
- correlate well with human judgement;
- be reliable and exhibit as little variance as possible;
- be cheap to set up and apply;
- be automatic.

4 Metrics for Performance Evaluation

Currently there is no standard for ontology-based information extraction (OBIE) because it is a relatively new area of research, although there are several well-established metrics for evaluation of traditional IE systems. The most common metrics are those defined by MUC [2] (Precision/Recall/F-measure) and ACE [1] (cost-based measure based on error rate). The needs of ontology-based information extraction metrics are rather different, however, because traditional methods are binary rather than scalar. This means that they assess an answer as correct or incorrect (occasionally allowing for partial correctness which is generally allocated a ”half-score”). Ontology-based systems should, however, be evaluated in a scalar way, in order to allow for different degrees of correctness. For example, classifying ”John Smith” as a Location rather than a Person is clearly wrong, but classifying him more generally as a Person (when he should have been classified more specifically as a Lecturer) is clearly less wrong than classifying him as a Location. Similarly misclassification at a more fine-grained level is clearly less wrong than misclassification at a more general level (e.g. classifying as a Research Assistant rather than as a Lecturer is less wrong than classifying him as a Location rather than a Person). A scalar method of evaluation allows the score to be based on the position of the response in the ontology and its closeness to the correct position in the ontology, and thus allows us to weight the score accordingly.

The CBE (Cost-Based Evaluation) model [13], which stems from the field of economics, is superior to Precision and Recall in some aspects, because it allows multi-dimensional evaluation, where a single score is not generated, but instead the evaluation is carried out simultaneously along several axes. This model is designed specifically for different applications or different users, who might have diverging requirements of a system. For example, one user might be more concerned with Precision than Recall, another user might be more concerned about getting particular types of entities right, and not so concerned about other types, and another user might be more concerned with the fact that even getting something partially right is important. Therefore a cost-based model is useful, particularly in industrial rather than research settings, because it enables the parameters to be modified according to the particular evaluation or task.

Multi-dimensional evaluation has also been applied to several existing systems. For example, Olsson et al. [10] evaluate the performance of protein name taggers in this way to overcome the limitations of Precision and Recall being too inflexible, proposing additional measures such as Sloppy, Left Boundary and Right Boundary to cater for responses which overlap the Key annotations. The GATE evaluation tools [5] provide something similar, where partially correct answers can be given a half weight (Aver-

age), counted as correct (Lenient) or counted as incorrect (Strict).

However, if a fully-fledged CBE model were to be adopted as a standard for ontology population evaluation, we would have to devise some simple and heuristic method of weight assignment, or in any case the creation of a generic set of weights that could be used as a default. Also, we would need some scoring tool, with the ability to be adapted easily by the user to reflect changes to the weights. Although the CBE model guarantees the most flexible application of various evaluation metrics, we have opted for a more simple version where we only take two dimensions into acccount: the Precision/Recall metric, and the semantic distance between key (gold standard) and response (system output) concepts in terms of a given ontology (similar to TRUCKS [9] and Learning Accuracy [6]). This method measures how well a particular text item has been classified.

5 Augmented Precision and Recall

In this section, we propose a new metric for performance evaluation, which we call Augmented Precision and Recall. This aims to preserve the useful properties of the standard Precision and Recall scoring metrics, but combines them with a cost-based component. It is based largely on Learning Accuracy (LA), but resolves one flaw: namely that LA does not take into account the depth of the key concept in the hierarchy, considering essentially only the height of the MSCA (Most Specific Common Abstraction) and the distance from the response to the MSCA. This means that however far away the key is from the MSCA, the metric will give the same outcome. We therefore propose a more balanced distance metric, which we call BDM. This uses the following measurements:

- MSCA: most specific concept common to the key and response paths
- CP: shortest path from root concept to MSCA
- DPR: shortest path from MSCA to response concept
- DPK: shortest path from MSCA to key concept

Each of these measurements needs to be normalised with respect to the average length of the chains in which key and response concepts occur. This will make the penalty that is computed in terms of node traversal relative to the semantic density of the chains [3]. For this purpose we have created the following normalisations:

- n1: the average length of the set of chains containing the key or the response concept, computed from the root concept.
- n2: the average length of all the chains containing the key concept, computed from the root concept.
- n3: the average length of all the chains containing the response concept, computed from the root concept.

The complete BDM formula is as follows:

$$BDM = \frac{CP/n1}{CP/n1 + DPK/n2 + DPR/n3} \quad (1)$$

This measure takes the relative specificity of the taxonomic positions of the key and response into account in the score, but it does not distinguish between the specificity of the key concept on the one hand, and the specificity of the response concept on the other. For instance, the key can be a specific concept (e.g. 'car'), whereas the response can be a general concept (e.g. 'relation'), or vice versa.

Essentially, the measure provides a score somewhere between 0 and 1 for the comparison of key and response concepts with respect to a given ontology. If a concept is missing or spurious, BDM is not calculated since there is no MSCA. If the key and response concepts are identical, the score is 1 (as with Precision and Recall). Overall, in case of an ontological mismatch, this method provides an indication of how serious the error is, and weights it accordingly.

We can now combine the BDM scores for each instance in the corpus, to produce Augmented Precision, Recall and F-measure scores for the annotated corpus. We differentiate this from traditional Precision and Recall due to the fact that it considers weighted semantic distance in addition to a binary notion of correctness.

$$BDM = \sum_{i=\{1...n\}} BDM_i \; and \; BDM_i = \frac{CP_i/n1_i}{CP_i/n1_i + DPK_i/n2_i + DPR_i/n3_i} \quad (2)$$

Augmented Precision (AP) and Recall (AR) for the corpus are then calculated as follows:

$$AP = \frac{BDM}{n + Spurious} \; and \; AR = \frac{BDM}{n + Missing} \quad (3)$$

while F-measure is calculated from Augmented Precision and Recall as:

$$F - measure = \frac{AP * AR}{0.5 * (AP + AR)} \quad (4)$$

6 Evaluation Prodecure

In order to enable the application and evaluation of the evaluation algorithms proposed in the previous section, we need an ontology and a text corpus that is semantically annotated with concepts from the ontology. In general, this method works if both resources are available for a particular conceptualisation, expressed in the ontology, and corresponding text annotations. A restriction on the nature of the ontology is that it must include hierarchical chains. For the evaluation matrix to be effective it cannot be just a set of named entities with no taxonomic embedding. If named entities are used as the only evaluation criterion, a binary metric with standard Precision and Recall suffices, i.e. the evaluation is in that case based on (partial) matching, missing annotations and false positives. To evaluate conceptual matching with respect to an ontology, we require a more complex evaluation mechanism, and have therefore chosen Augmented Precision and Recall. This metric also fulfils important evaluation criteria such as ease of implementation, simplicity, coverage, scalability, repeatability, and ease of comprehension of the results. Note that here we mean criteria for the evaluation metric itself, not criteria for the systems to be evaluated.

7 Conclusions

In this paper we have discussed the importance of benchmarking for semantic annotation tools, particularly with respect to the biomedical domain where such tools are vitally important for the development of ontology-based applications and knowledge extraction. We have described a suite of evaluation criteria which measures not just performance but also issues such as usability, scalability and interoperability, and we have proposed a new metric for ontology-based evaluation which takes into account the hierarchical structure of ontologies and its consequences for evaluation. The new evaluation metrics proposed will be made available in the GATE framework [5] and experiments with the Gene Ontology and other ontologies and corpora are currently underway.

Acknowledgements

This work is partially supported by the EU-funded Knowledge Web network of excellence ((IST-2004-507482) and SEKT project (IST-2004-506826).

References

[1] ACE. *Annotation Guidelines for Entity Detection and Tracking (EDT)*, Feb 2004. Available at http://www.ldc.upenn.edu/Projects/ACE/.

[2] Advanced Research Projects Agency. *Proceedings of the Fifth Message Understanding Conference (MUC-5)*. Morgan Kaufmann, California, 1993.

[3] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proc. of 16th International Conference on Computational Linguistics*, volume 1, pages 16–23, Copenhagen, Denmark, 1996.

[4] 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*. IOS Press, Amsterdam, 2003.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

[6] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *Proc. of 15th National Conference on Artificial Intelligence (AAAI-98)*, pages 524–531, Menlo Park, CA, 1998. MIT Press.

[7] M. King. Living up to standards. In *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing*, Budapest, Hungary, 2003.

[8] D. Maynard, M. Yankova, A. Kourakis, and A. Kokossis. Ontology-based information extraction for market monitoring and technology watch. In *ESWC Workshop "End User Apects of the Semantic Web")*, Heraklion, Crete, 2005.

[9] D.G. Maynard and S. Ananiadou. Term extraction using a similarity-based approach. In *Recent Advances in Computational Terminology*. John Benjamins, 1999.

[10] F. Olsson, G. Eriksson, K. Franzn, L. Asker, and P. Lidn. Notions of Correctness when Evaluating Protein Name Taggers. In *Proceedings of COLING 2002*, Taipei, Taiwan, 2002.

[11] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM – Semantic Annotation Platform. *Natural Language Engineering*, 2004.

[12] P.W.Lord, R.D. Stevens, A. Brass, and C.A.Goble. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–83, 2003.

[13] Peter Sassone. Cost-benefit methodology for office systems. *ACM Transactions on Office Information Systems*, 5(3):273–289, 1987.

[14] W. Siberski. Semantic web framework requirements analysis. Technical Report D1.2.2, KnowledgeWeb Deliverable, 2005.

[15] H. Wache, L. Serafino, and R. Garcia Castro. Scalability - state of the art. Technical Report D2.1.1, KnowledgeWeb Deliverable, 2005.