ROSeAnn: Reconciling Opinions of Semantic Annotators

Luying Chen, Stefano Ortona, Giorgio Orsi, and Michael Benedikt
Oxford University, UK
name.surname@cs.ox.ac.uk

ABSTRACT

Named entity extractors can be used to enrich both text and Web documents with *semantic annotations*. While originally focused on a few standard entity types, the ecosystem of annotators is becoming increasingly diverse, with recognition capabilities ranging from generic to specialised entity types. Both the overlap and the diversity in annotator vocabularies motivate the need for *managing and integrating semantic annotations*: allowing users to see the results of multiple annotations and to merge them into a unified solution.

We demonstrate ROSEANN, a system for the management of semantic annotations. ROSEANN provides users with a unified view over the opinion of multiple independent annotators both on text and Web documents. It allows users to understand and reconcile conflicts between annotations via *ontology-aware aggregation*. ROSEANN incorporates both supervised aggregation, appropriate when representative training data is available, and an unsupervised method based on the notion of weighted-repair.

1. INTRODUCTION

A growing number of resources are available for recognising named entities in documents (e.g. London, the King of Spain) and to link them to particular semantic entity types (e.g. politicians, governmental organizations) generating *semantic annotations*. While originally focused on a few standard top-level types such as people, locations, and organizations, the ecosystem of annotators is becoming increasingly diverse. Modern annotators support broad vocabularies consisting of both common-knowledge, e.g., persons, and specialised entity types, e.g., proteins [5]. Annotations play an important role, e.g., in semantic search engines, information extraction, and for the automated production of linked data. Annotation functionality is frequently obtained via online "black-box" services (e.g. OPENCALAIS, ZEMANTA). This makes it very easy for users to embed annotation capabilities in their applications. It also creates challenging problems, including judging the quality of annotations and reconciling disagreeing opinions about an entity.

Consider the example in Figure 1. Here we see the variations in quality within annotators, as well as an idea of several flavors of clash in annotator opinions. The token "*Japanese*" is labeled as an *Organisation* by WIKIMETA, as a *Language* by DBPEDIA SPOTLIGHT, and as a *Country* by EXTRACTIV– clearly these outputs are incompatible, since these three entity types have mutually disjoint meaning in a given context. On the other hand, some annotators claiming knowledge of potential entity types for tokens did not annotate with these types – this could also be an indication of the semantics of a given token in the snippet. As an example, OPENCALAIS and ZEMANTA know the entity type *Organisation* but do not provide an annotation for the token, which could be seen as evidence that it is not an *Organisation*. Moreover, EXTRACTIV knows the concept of *Nationality* – a reasonable entity type for *Japanese* – but does not annotate the token as such.

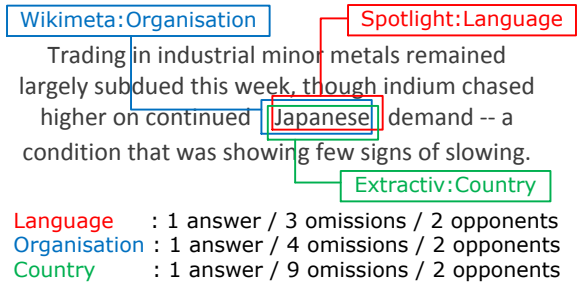

Figure 1: Conflicting and re-enforcing annotations.

We demonstrate ROSEANN, a system for integration and reconciliation of semantic annotations provided by multiple independent annotators. ROSEANN concurrently annotates text or Web documents with multiple independent annotators, and then presents a unified view as a single annotated document. It allows browsing and filtering of annotations, using a number of filtering mechanisms, and allows the inspection of annotations which are conflicting or supporting each other. Most importantly, it allows for *aggregation* of annotations, returning a logically-consistent set of annotations w.r.t. a background ontology. We evaluate ROSEANN on well-known corpora, showing significant improvement w.r.t. individual annotators and state-of-the-art aggregators such as FOX [6] and NERD [9].

Architecture. The ROSEANN architecture is shown in Figure 2. A controller interacts with two kinds of services, *annotators* and *aggregators*. Annotators independently locate named entities within text or Web documents, while aggregators take as input the opinions of multiple annotators, along with an ontology, and return a logically-consistent merged annotation. ROSEANN currently supports eleven annotators, namely: OPENCALAIS, EXTRACTIV, DBPEDIA SPOTLIGHT, ALCHEMYAPI, ZEMANTA, LUPEDIA, WIKIMETA, SAPLO, YAHOOYQL, STANFORDNER, and

Proceedings of the VLDB Endowment, Vol. 6, No. 12

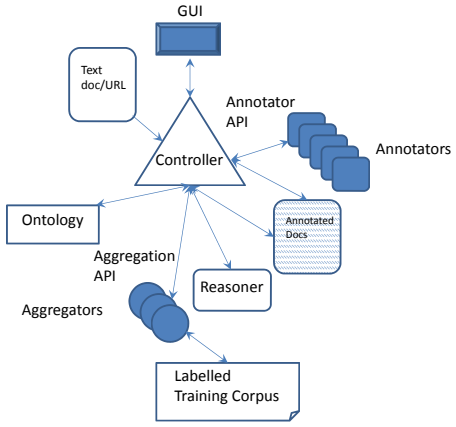

Figure 2: ROSEANN architecture.

NETAGGER. ROSEANN currently supports two aggregation methods: one *supervised* and one *unsupervised* and uses the reconciled entity types to give links to external LOD resources. ROSEANN interfaces with the background ontology and reasoning services via an OWLIM SPARQL endpoint.

Organization: Section 2 describes our aggregation algorithms. Section 3 discusses the implementation and experimental evaluation, while Section 4 gives the details of the demonstration.

2. RECONCILING ANNOTATIONS

ROSEANN performs the aggregation of individual annotators on the basis of an ontology mapping and an aggregation algorithm. The ontology mapping has been created by taking the union of all entity types recognised by the annotators and by manually aligning them. While creating the mapping, we took into account the documentation of the annotators, their behaviour on real documents, and other reference ontologies such as those provided by DBPedia [3], Freebase [4] and Schema.org. Moreover, ROSEANN automatically reports to users new entity types reported by an annotator, asking for an alignment with those in the current global ontology.

ROSEANN interfaces with eleven online (e.g., OPENCALAIS) and standalone (e.g., STANFORDNER) annotators and supports both supervised and unsupervised aggregation. Although one might expect that conflicts between these annotators are rare, our evaluation (Section 3) shows that they are indeed extremely frequent and justify an investigation of reconciliation techniques.

The supervised aggregator is based on a *Maximal Entropy Markov Model* (MEMM), a machine-learning-based method frequently used by individual annotators [7]. As in traditional Markov Models, MEMM captures patterns of sequential behaviour in a probabilistic transducer; it has a set of states, an input and output alphabet, and probabilistic transition and emission functions that capture the system dynamics when a new input item is consumed. The inputs consist of sets of token-level features. As with other maximal entropy models, we allow for overlapping feature functions, without requiring strong independence assumptions. MEMM has been trained to recognise entity types from the mapping ontology.

In the absence of dependable training data, ROSEANN provides a fully unsupervised alternative to MEMM based on the notion of *weighted repair*. This notion is a weighted extension of the approach adopted for consistent query answering over inconsistent knowledge bases [10], where the weighting represents, roughly speaking, the amount of support or opposition that is accorded to a given repair action.

Consider again the example Figure 1 where a span $\hat{s}$ is tagged by several annotators. Entity types can be identified with atomic propositions, and the ontology relationships can be considered as propositional constraints – e.g. if C and D are disjoint entity types, our logical theory includes the constraint $C \rightarrow \neg D$. Thus we can translate the ontology Ω to a propositional theory T_Ω. We say that an annotator *supports* an entity type C if it tags the span $\hat{s}$ with (a subclass of) C. Dually, we say that an annotator *opposes* when it tags $\hat{s}$ with a class disjoint from C or when an annotator fails to tag $\hat{s}$ with (a superclass of) C that is in its vocabulary (opposition via omission). We associate to each identified type C an integer value $\mathsf{AtomicScore}(C)$, representing the degree of support for or opposition to C by annotators. The general form of our scoring function is:

$$\begin{aligned}\mathsf{AtomicScore}(C) = \Sigma_{\mathsf{A}\in\mathsf{Anns}} \\ \Sigma_{D\sqsubseteq C\in\Omega}\mathsf{SupportWeight}_{\mathsf{A},D}\cdot\mathsf{Support}(\mathsf{A},D) \\ -\Sigma_{D\sqcap C=\bot\in\Omega}\mathsf{SupportWeight}_{\mathsf{A},D}\cdot\mathsf{Support}(\mathsf{A},D) \\ -\Sigma_{C\sqsubseteq D\in\Omega}\mathsf{OmitWeight}_{\mathsf{A},D}\cdot\mathsf{Omit}(\mathsf{A},D)\end{aligned}$$

Above, Anns denotes the set of annotations, $D \sqsubseteq C \in \Omega$ indicates that from the rules of ontology Ω one can prove D is a subclass of C, and $D \sqcap C = \bot \in \Omega$ indicates that Ω implies disjointness of D and C. $\mathsf{Support}(\mathsf{A},D)$ is 1 if annotator A tags the span s with D, and is 0 otherwise. $\mathsf{Omit}(\mathsf{A},D)$ is 1 iff A has D in its vocabulary, but failed to tag span s with D. $\mathsf{SupportWeight}_{\mathsf{A},D}$ and $\mathsf{OmitWeight}_{\mathsf{A},D}$ are non-negative $[0,1]$-valued weights that indicate how much weight the tagging of A with D or the omission of D by A should have.

Given the atomic scores, a boolean combination of entity types that is consistent with the ontology is computed. Our *weighted repair* (WR) algorithm first takes the union of all entity types returned by any annotator, which can be considered as a conjunction of entity types σ_{Init}. A repair operation Op is either a deletion of an entity type occurring as a conjunct within σ_{Init} or an insertion of a entity type that is absent from σ_{Init}. The application of Op to σ_{Init} produces a new formula. For a deletion of class C, it is formed by removing every conjunct corresponding to a subclass of C while adding the negation of C, while for an insertion it is formed by conjoining with a proposition corresponding to C. A set of repairs is *internally-consistent* if no two operations conflict, e.g., we do not delete a class C and also insert a subclass of C. For an internally-consistent set of repairs $S = \{\mathsf{Op}_1 \ldots \mathsf{Op}_n\}$, the application on σ_{Init}, denoted $S(\sigma_{\mathsf{Init}})$ is defined as the result of applying the Op_i in any order. A repair set is *non-redundant* if we do not delete or insert two entity types in a subclass relation. A *solution* is an internally-consistent, non-redundant repair set S such that $S(\sigma_{\mathsf{Init}})$ is consistent with T_Ω. Our goal is to find a solution with maximal *aggregate score* among all solutions, where the aggregate score of a solution S is:

$$\Sigma_{\mathsf{Ins}(C)\in S}\mathsf{AtomicScore}(C) - \Sigma_{\mathsf{Del}(C)\in S}\mathsf{AtomicScore}(C)$$

That is, an operation that deletes an entity type C incurs the penalty $\mathsf{AtomicScore}(C)$, while an insertion of an entity type C incurs the negative of $\mathsf{AtomicScore}(C)$ as a penalty.

Since multiple repairs can achieve the maximal score, we impose ranking criteria: 1. Given two solutions with the same score and different numbers of repairs, we prefer the smaller one. 2. Given solution $S_1 = S' \cup \{\mathsf{Ins}(C_1)\}$ $S_2 = S' \cup \{\mathsf{Ins}(C_2)\}$, with C_2 a subclass of C_1, we prefer S_2, i.e., the one that inserts more specific classes.

ROSEANN computes the optimal solution by reducing the above optimization problem to integer linear programming (ILP). With reference to the example of Figure 1, WR returns as output a solution with a single *Language* annotation since it is logically consistent and also with less opposition from the other annotators. On the other hand, MEMM returns a solution with an annotation of type *Nationality*. MEMM learns from the training set that there is a correlation between the annotations *Country* and

Language provided by EXTRACTIV and DBPEDIA SPOTLIGHT, and the entity type *Nationality*. For a detailed description of WR and MEMM, we refer the reader to the technical report available at: http://diadem.cs.ox.ac.uk/roseann.

3. EVALUATION

Datasets. We evaluated ROSEANN on four different benchmark corpora. **(i)** The MUC7 dataset [1], consisting of 300 newswire articles, annotated with standard high-level entity types such as Person, Organization, Location and Money; **(ii)** the Reuters corpus (RCV1) [2], a general information retrieval corpus of English newswire documents; **(iii)** the corpus used by the FOX [6] entity extractor and consisting of 100 snippets of text from newswires; **(iv)** the corpus used by the Illinois NETAGGER [8] entity extractor and consisting of text sourced from 20 web pages mostly about academics.

For MUC7 and the FOX and NETAGGER corpora we used the original gold standard provided by these benchmarks. In the case of the Reuters corpus, we looked at five of the most common Reuters topics – *Entertainment&Sports*, *Financial&Economics*, *Healthcare&Social*, *Products* and *Tourism&Travel* – and randomly sampled 250 documents from the 810k available in the corpus, distributing evenly over the topics. For this sub-corpus of Reuters, we manually annotated the documents by using the most specific entity types from the global vocabulary of the mapping ontology. For the FOX and NETAGGER both the original documents and gold-standard annotations are available online at http://diadem.cs.ox.ac.uk/roseann. Due to copyright reasons, for the Reuters and MUC7 we made available from the same website only the gold-standard annotations with pointers to the original Reuters and MUC7 documents via the corresponding document ID.

Precision and Recall. We measure PRECISION and RECALL in a way that is *ontology-aware*: for example, given an ontology Ω, if an annotator declares a given span to be a Person while our gold standard indicates that it is an Artist, then this annotator will be eventually penalized in recall for Artist (since it had a miss), but not, e.g., in precision for Person (since it can be inferred via Artist).

More precisely, we define the precision of an annotator AN for an entity type C as:

$$\text{PRECISION}_{\Omega}(C) = \frac{|Inst_{\text{AN}}(C^+) \cap Inst_{GS}(C^+)|}{|Inst_{\text{AN}}(C^+)|}$$

where $Inst_{\text{AN}}(C^+)$ denotes all instances in the test set annotated as (a subclass of) C by AN, and $Inst_{GS}(C^+)$ denotes all instances in the test set determined to be (a subclass of C) in the gold standard. In computing the intersection, we use a "relaxed" span matching, which requires only that the spans overlap.

We define the recall of an annotator AN for an entity type C in an analogous way, again using the "relaxed" notion of span-matching for the intersection:

$$\text{RECALL}_{\Omega}(C) = \frac{|Inst_{\text{AN}}(C^+) \cap Inst_{GS}(C^+)|}{|Inst_{GS}(C^+)|}$$

Based on the extended definitions of precision and recall, the F_1-SCORE for an entity type C is defined in the standard way.

Results. To test the need for reconciliation, we collected data about the extent and distribution of annotator conflict. We consider two forms of conflicts: *basic conflicts* occur when one annotator annotates the span with an entity type C, and another annotator which has (a superclass of) C in its vocabulary fails to annotate the same span with it. For annotators with low recall, a basic conflict may be a weak indicator of a wrong annotation. Thus we also consider *strong conflicts*, which denote situations when two annotators annotate the same span with entity types C and C', where C and C' are disjoint. For instance, on the MUC7 corpus we detected 36,756 basic conflicts and 3,501 strong conflicts, while on the Reuters corpus we detected 21,639 basic conflicts and 2,937 strong conflicts. These numbers show also that annotation reconciliation is essential.

The performance of the aggregation via WR and MEMM has been first compared against each individual annotator. For the evaluation we considered only the entity types that are common between the aggregator and the given annotator. Since entity types are unevenly represented in the corpora, we computed *micro* and *macro* averages over precision, recall and F_1-score. The *micro* average first computes the sums of true-positive, false negatives, and false positives for each entity type and then computes an overall precision, recall, and F_1-score using these quantities. The *macro-average* is a straightforward average of precision, recall, and F_1-score computed over each entity type.

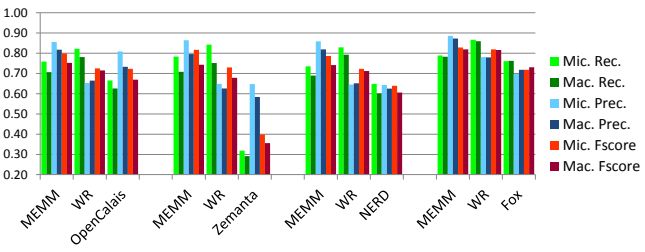

Figure 3: ROSeAnn evaluation.

Figure 3 summarizes the most relevant results of the evaluation on all four corpora. Both WR and MEMM outperform each individual annotator w.r.t. F_1-score. In particular, Figure 3 shows the best and worst individual annotators. OPENCALAIS is the annotator who is closest in performance to ROSEANN. It is worth mentioning that OPENCALAIS performs about 0.02% better than WR and about 3% better than MEMM on the Reuters corpus alone. On the other hand, its vocabulary represents about only 18% of all entity types in the gold standard. ROSEANN achieves the best performance against ZEMANTA with an improvement of about 40% via MEMM.

We then compared ROSEANN against two state-of-the-art annotation aggregators: FOX and NERD. When comparing against an aggregator, we considered the same individual annotators and the same entity types supported by the given aggregator. This enables a fair evaluation of the aggregation algorithm alone without being affected by the choice of individual annotators. The results show that both WR and MEMM outperform both FOX and NERD in average of 10% and 17% respectively. More detailed performance tables and charts are available online at: http://diadem.cs.ox.ac.uk/roseann/EvaluationResults.zip

4. DEMONSTRATION WALK-THROUGH

The demonstration showcases ROSEANN's semantic annotation and reconciliation capabilities. ROSEANN provides a graphical interface (Figure 4) enabling users to load both static text and live web documents into the tool. Web navigation is provided by driving a Firefox web browser via Selenium WebDriver[1].

The main menu bar on the top of the GUI (**A**) supports document loading, annotation, ROSEANN configuration, as well as the possibility to save the annotated documents and to browse the entity types of the mapping ontology via the ROSEANN SPARQL Endpoint[2]. Annotated documents are accessible from the left-hand side

[1] http://docs.seleniumhq.org/.

[2] http://163.1.88.38:8081/openrdf-workbench/.

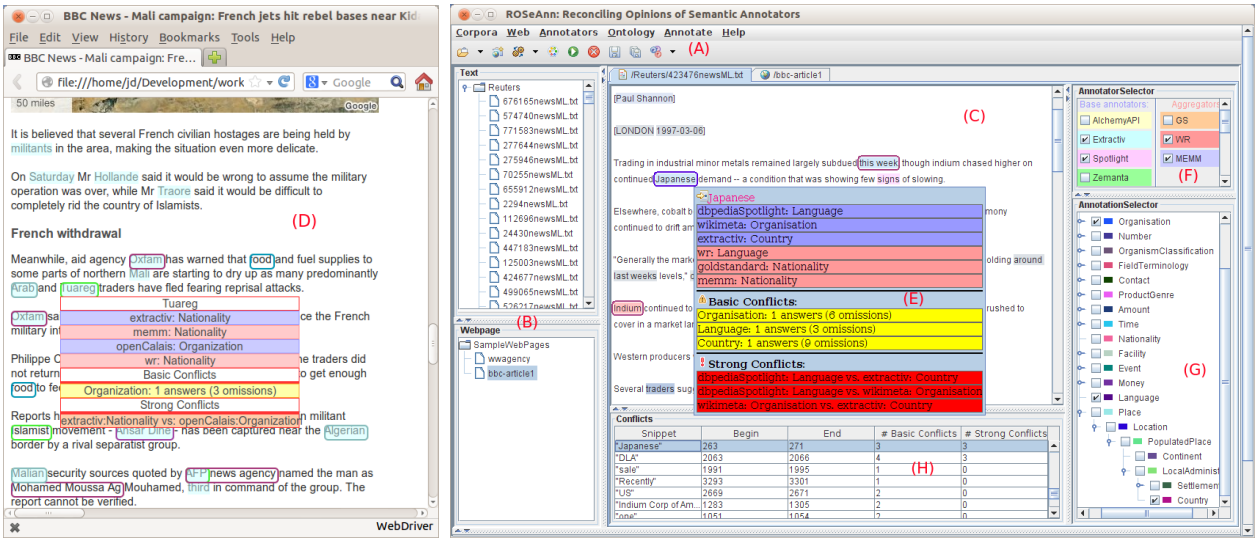

Figure 4: ROSeAnn GUI components.

of the GUI (**B**), together with documents coming from the Reuters and MUC7 corpora, which are used as pre-loaded benchmarks for our evaluation.

Textual documents are visualized in the main text area of the tool (**C**), while web documents are visualised in an independent browser window (**D**). In both cases, the user can interact with the document and the web browser before starting the annotation process.

After a document has been annotated, ROSEANN highlights the recognised entities in the main text area or in the web browser. The highlighting consists of a colored border around the identified entities representing an entity type in the ontology, and a colored background representing the annotator or aggregator recognising that particular entity. For documents coming from the Reuters and MUC7 corpora, we also provide the gold-standard annotations. By hovering over the highlighted entities, ROSEANN provides the list of opinions for all annotators and aggregators specifying which annotator contributed a given entity type, together with the basic and strong conflicts (**E**). A different background on the tooltip is used to distinguish the opinions of annotators from those of the aggregators. When an annotator provides also links to LOD, e.g., to DBPedia, ROSEANN makes available those anchors to the user in the tooltip.

On the right-hand-side of the GUI we list all annotators (**F**) that identified at least one entity in the current document and the identified entity types organised into a hierarchy (**G**) that corresponds to the structure of our mapping ontology. The user can decide which annotators, aggregators and entity types to visualise in the main text-area or in the browser.

At the bottom of the GUI we provide a table listing all conflicts generated by the annotators in the given document (**H**). In particular, we report (from the left to the right in the table) the text snippet involved in the conflict, the start and end offset of the text span in the document, and the number of basic and strong conflicts occurring on that span. After selecting a row in the conflict table, ROSEANN blinks the span involved in the conflicts in the text area or in the browser.

In the demonstration we first showcase example text and web documents from the newswire domain and, in particular, from the Reuters and MUC7 corpora and from news websites such as the BBC, Reuters and New York Times. We will show the most common situations arising conflicts among annotations and the resolution performed by WR and MEMM. The demonstration will then proceed with live websites from different domains and user-provided documents.

5. REFERENCES

[1] MUC7 http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2001T02.

[2] Reuters http://about.reuters.com/researchandstandards/corpus/index.asp.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[5] R. Carreira, S. Carneiro, R. Pereira, M. Rocha, I. Rocha, E. C. Ferreira, and A. Lourenço. Semantic annotation of biological concepts interplaying microbial cellular responses. *BMC bioinformatics*, 12(1):460, 2011.

[6] FOX. http://ontowiki.net/Projects/FOX?v=4e5.

[7] A. McCallum, D. Freitag, and F. C. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, pages 591–598, 2000.

[8] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on CoNLL*, pages 147–155, 2009.

[9] G. Rizzo and R. Troncy. Nerd: A framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the EACL 13th Conference*, pages 73–76, 2012.

[10] R. Rosati. On the complexity of dealing with inconsistency in description logic ontologies. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pages 1057–1062, 2011.