Time to evaluate: Targeting Annotation Tools

Peyman Sazedj[1] and H. Sofia Pinto[1]

Instituto Superior Técnico,
Departamento de Eng. Informática,
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
psaz@ist.utl.pt, sofia.pinto@dei.ist.utl.pt

Abstract. One of the problems newcomers to the annotation area face is which tool to choose. In order to compare available tools one must evaluate them according to an evaluation framework. In this paper we propose an evaluation framework, that introduces general tool-focused criteria – interface and general – and annotation-focused criteria - metadata and procedure. We describe how the evaluation of a set of tools was performed and the results of such a comparison. Our results may not only be useful to newcomers, but also to experts who may use them to seize new opportunities for development.

1 Introduction and Motivation

The term annotation either designates *metadata*, or a *procedure* (that produces metadata). Based on this definition, an annotation tool is a tool that either allows to manipulate annotation metadata, that operates an annotation procedure, or both. Annotation metadata can have a textual, ontological or linguistic nature. Annotations are associated to resources, which can take several forms: web pages, images, video. If we take these dimensions there is already a considerable wealth of tools that fall into the annotation tool category. In our case we are interested in ontological annotation tools that deal with web pages.

Moreover, there are several other dimensions that characterize and distinguish annotation tools. Therefore, one of the problems newcomers to the annotation area face is which tool to choose. Although some initial attempts to describe some of the available tools have been made [1], these attempts aimed only to list and describe some functionality issues (languages offered, kind of annotation procedure supported - manual or automatic -, etc.) However, due to the vast array of different features provided by these tools it would be interesting to rank them according to a given set of features.

In order to compare available tools one must evaluate them according to an evaluation framework. In this paper we propose an evaluation framework, section 3 that introduces domain specific and domain independent criteria. These criteria range from general tool-focused criteria – interface and general – to annotation-focused criteria - metadata and procedure. We describe how the evaluation of a set of tools was performed, section 4 and the results of such a comparison, section 5 and take some initial conclusions from our findings, section 6.

2 Choosing the Tool Set

In order to select among existing annotation tools, we chose two simple selection criteria. The first criterion captures the requirement of providing explicit formal meaning to annotations. From among the technologies we have seen around so far, only ontology-based tools assign semantics to their data. Thus, we were only concerned with the evaluation of **ontology-based annotation tools**. The diversity of resources available on the web, lead us to impose a second criterion. Since the largest part of the world wide web is composed of web pages, and since web pages are the main medium for hosting other types of resources, we were mostly interested in the **annotation of web pages**.

Based on these two criteria, a set of 5 tools was selected: `Kim Plugin 1.05` [2] (Sirma Inc.), `Melita 2.0` [3] (University of Sheffield), `MnM 2.1` [4] (KMI), `Ontomat 0.8` [5] (AIFB) and `C-Pankow` [6] (AIFB).[1] Even though Ontomat is supposed to support automatic annotation, due to technical reasons, we could only operate it in manual mode. Melita and MnM are semi-automatic annotation tools based on the Amilcare Information Extraction Engine [7]. Kim and C-Pankow are fully automatic and unsupervised tools. Kim is a semantic information platform, where only the Kim annotation plugin and its annotations were evaluated. With regard to C-Pankow, we evaluated its web-based implementation[2].

3 Evaluation Framework

We propose an evaluation framework for the evaluation of annotation tools, based on a set of well-defined criteria. Our hope is to define each criterion as clearly as possible, encouraging a rigorous and fair evaluation. For that purpose, a metric is associated with each criterion, quantifying its measurement as accurately as possible. Since perfect quantification rather exceeds our scope, our metrics are mostly significant to a relative extent. In other words, the final quantification of a tool can only be meaningful when compared to the quantification of another tool, evaluated by the same framework according to the same set of criteria. Moreover, we try to identify as many criteria as possible, representing the diverse issues dealt with by the area of semantic annotation to a fair extent.

We propose a set of 20 criteria which are classified into four dimensions: general, interface, metadata and procedure. We argue that any criterion is either *domain-independent* or *domain-specific*. In turn, domain-independent criteria are divided into *interface* and *general* criteria, whereas *domain-specific* criteria are divided into *metadata* and *procedure* criteria. Even though some of our interface criteria may apparently seem to have domain-specific metrics, the criteria themselves are essentially domain independent and their metrics can be adapted to any domain of choice.

Based on their metrics, our criteria may be classified into three types:

[1] Smore 5.0 was not evaluated due to lack of support of annotation metadata.

[2] http://km.aifb.uni-karlsruhe.de/services/pankow/annotation/

1. Feature-based criteria
2. Set-dependent criteria
3. Set-independent criteria

Feature-based criteria are those which verify the existence of specific features or functionalities. Most of our domain-specific criteria are of this type. *Set-dependent* criteria are those which depend on some property inherent to the whole set of tools under evaluation. In other words, the same tool may score differently if evaluated within distinct sets of tools. *Set-independent* criteria are based on mathematical metrics which are independent of the tool set.

In table 1 we present a list of all criteria. Feature-based criteria are indicated with (f), while set-dependent criteria are marked with (d). Criteria with no marks are set-independent. The asterisk indicates that a criterion is only applicable to tools with support for automatic annotation.

Interface	*Metadata*	*Procedure*	*General*
Self-documentation	Association (f)	Expressiveness (f)	Documentation
Simplicity	Flexibility (f)	Input (f)	Scalability (f)
Timeliness	Heterogeneity (f)	Output (f)	Stability
Usability (d)	Integrity (f)	Precision*	
	Interoperability (d)	Recall*	
	Scope (f)	Reliability*	
		Speed*	

Table 1. All criteria, classified into four dimensions.

Since we classified our criteria into one of three types, we present a detailed example of each type within the remainder of this section. Table 2 summarizes our domain-specific criteria. Due to lack of space we omit general and interface criteria tables.

Heterogeneity is a feature-based criterion, defined as the quality of being able to combine diverse types of knowledge. The criterion defines the following features: (1) The tool permits the simultaneous loading of multiple Ontologies; (2) The tool supports the annotation of the same "unit" with multiple concepts. The metric is simply based on assigning a score of one point per feature. Since the criterion defines two features, a tool may score zero, one or two points depending on whether it fulfills none, one or both features.

Interoperability is a set-dependent criterion, defined as the ability to exchange and use information among different tools. Its metric is defined as:

$$\left(\frac{F_r}{F_{r.max}} + \frac{F_w}{F_{w.max}}\right) * \frac{1}{2} * 100\%$$

Consider a set of tools T. A tool $t \in T$ is considered inter-operable with a tool $t' \in T$ where $t \neq t'$, if it can write in a format which t' can read and if it can read the formats which t' can write. In order to be 100% interoperable with all tools, a tool t needs to be able to read all formats which the tools $\forall t' \in T, t \neq t'$ can write and needs to be able to write a subset of formats F',

Name	*Short Definition*	*Range*	*Metric*
Association	The way an annotation is associated with the annotated resource.	[0,2]	1 point per feature: (1) Tool supports external annotations. (2) Tool supports internal annotations.
Flexibility	The quality of being adaptable or variable.	[0,3]	1 point per feature: (1) It is possible to delete previously created annotations. (2) There is a component for editing ontologies. (3) It is possible to define a custom namespace to be used with the annotations.
Integrity	The capacity of ensuring the soundness and integrity of an annotation.	[0,3]	1 point per feature: (1) Tool verifies the domain and target constrains of a relation. (2) Tool certifies the correctness of an association over time. (3) Tool verifies the existence of an association.
Scope	The scope of an annotation corresponds to the parts of a resource to which the annotation applies.	[0,3]	1 point per feature: (1) Annotate the minimum unit of information. (2) Annotate any multiple of that minimum unit. (3) Annotate the resource as a whole, instead of the multiple of all units.
Expressiveness	The capacity of fully conveying the meaning of the text through annotations.	[0,5]	1 point per feature: (1) Tool supports annotation with concepts. (2) Tool supports annotation with relations. (3) Tool supports the instantiation of concepts. (4) Tool supports the instantiation of relations. (5) Tool associates mark-up with instances.
Input	The different types of resources which are supported as input to the annotation procedure.	[0,3]	1 point per feature: (1) Web page (2) Text document (3) Image
Output	The output of the annotation procedure. It is verified whether the tool writes annotations in a format usable on the Semantic Web.	[0,2]	1 point per feature: (1) Annotations are written in a formalism strictly conforming to a W3C standard (2) Annotations are unambiguously associated with ontological concepts
Precision	The precision of an algorithm is measured by the certainty of the algorithm in producing correct results.	[0,1]	$P = \frac{C}{T}$ C - Number of correct annotations T - Total number of annotations
Recall	The recall of an algorithm is measured by the percentage of correct solutions found while discarding incorrect solutions.	[0,1]	$R = \frac{C}{M}$ C - Number of correct annotations M - Ideal number of annotations
Reliability	The reliability of the algorithm in producing the expected results.	[0,1]	$R = \frac{2 * P * R}{P + R}$ P - Precision of the algorithm R - Recall of the algorithm
Speed	The speed of an annotation algorithm is the ratio of the number of annotations produced, by the time it took to produce the results.	[0,1]	$S = 1 - \frac{T}{T + A}$ A - Number of named entities analyzed for annotation T - Time it took to produce annotations

Table 2. Summarized definition of metadata and procedure criteria (annotation-specific criteria), excluding the Heterogeneity and Interoperability criteria which are explained within the main text of this article.

such that any tool t' is able to read at least one of the formats $f \in F'$. Based on this discussion, $F_{r.max}$ is the cardinality of the set of all formats which can be written by annotation tools different from the one under evaluation. $F_{w.max}$ is the cardinality of a minimum set F' of formats such that any tool (other than the one under evaluation) should be able to read one of the formats $f \in F'$. F_r is the number of different annotation formats the tool can read and F_w is the number of different formats the tool can write.

Simplicity is a set-independent criterion, defined as the quality of being simple, intuitive and easy to understand. Its metric is based on a set of sub-criteria where each sub-criterion is assigned a rating of 1 or 0, depending on whether the criterion applies or not:

$$S = L_c + E_f + C_l + C_o$$

The learning curve (L_c) designates the ease of learning the tool. If it is assigned a value of 1, then the tool can be learned within a few hours (typically less than 5), otherwise, the tool is more complex and eventually the help of an expert is needed in order to understand some of the provided features. By learning a tool, we mean not only how to produce annotations, but also how to work with most of the features provided. Thus, tools with many features typically have a higher learning curve. By efficiency (E_f) we designate the simplicity of producing annotations, once the tool has been learned. A heuristic for measuring efficiency is whether the tool allows to create a valid annotation, in less operations than most other tools. Even so, additional details have to be taken into account: Some tools provide error-checking while others don't. On the long term, tools that do provide error-checking may be more time saving than tools that don't, even though the latter may create annotations in less steps. We understand that this criterion is slightly subjective as it is. Clearness (C_l) determines how well a novice user gets along with simple tasks. This criterion is different from learning curve, because it is not concerned with understanding all the features and functionalities of the tool. Instead, the question is whether it is easy to locate a specific feature. For example, giving a novice user the task of opening an ontology from disk, the question is whether the interface is clear enough for him to easily perform this task. Finally, consistency (C_o) measures whether the tool is in conformance with actual standards and whether it is consistent with itself. For example, if there are different icons to perform the same operation, the tool is inconsistent. It also has to be taken into account whether other tools that perform similar operations, use the same icons or not. The same is true for the hierarchy and names of menus.

4 Evaluation Procedure

Having defined all criteria, the next step was to devise how the evaluation of the selected tools should be carried out. It was clear that not all tools could be evaluated according to all criteria. For example, our *precision*, *recall* and *reliability* criteria are only applicable to semi-automatic and automatic annotation tools and cannot be applied to manual annotation tools (Ontomat). MnM, Melita, Kim and C-Pankow were evaluated according to all criteria. A word must be said concerning the applicability of our *stability* criterion with regard to *remote* services, since both Kim and C-Pankow were tested as remote services. Let's suppose the invocation of a service sometimes fails. Due to lack of evidence, we cannot draw any conclusion regarding the cause of failure. The problem can either lie at the remote service, or at our internet connection, or elsewhere. Just in case a service never fails, we can assume that it is stable.

Although most criteria were of simple application, some criteria required detailed preparation. The *usability* criterion was measured by means of Jacob Nielsen's heuristic evaluation method [8], with the help of a team of three expert evaluators. All other criteria were measured by a single domain expert.

Tools with support for automatic annotation were tested against a set of corpora which were manually tagged. Within the following sections we describe the corpora and the ontologies which we used. We also describe an inter-annotation experiment which was carried out.

4.1 Corpora

The first corpus was the Planet Visits corpus, courtesy of the Knowledge Media Institute (KMI), a well known corpus, since it comes included with some annotation tools. We selected a set of 30 web pages, each containing one news article. The particularity of this corpus is that all news are about visits, either by someone from the KMI or by some external entity to the KMI. The documents of this corpus may be considered *unstructured* text.

Our second *unstructured* corpus was a smaller collection of news articles extracted from the Baha'i International Community world news service.[3] The corpus consists of three larger articles, with a total count of 1582 words.

Finally a third *tabular* corpus was created specifically for the evaluation of semi-automatic annotation tools, based on the observation that the performance of semi-automatic tools depends on the structure of documents. Since annotation with semi-automatic tools was not successful on the previous corpora, we created this *tabular* corpus hoping to obtain useful results. The corpus consists of a single table of 100 rows[4], and two columns, where the annotation task is as simple as classifying the words of the second column, with the corresponding ontological concepts of the first column. Concerning the two sample entries in table 3, `Siscog` would have to be annotated as an `Organization` and `Germany` as a `Location`.

Organization:	Siscog
Location:	Germany

Table 3. Small extract of the tabular corpus. The original corpus contains 100 rows.

[3] http://www.bahaiworldnews.org

[4] The corpus was later divided into two tables of 50 rows each, one for training and the other for testing.

4.2 Ontologies

After choosing the corpora, specific ontologies had to be designed for the annotation of each corpus. This task was a tedious one, due to the apparent lack of *interoperability* between tools. Among the tools with support for automatized annotation, KIM works with its own meta-ontology, C-Pankow and MnM support ontologies formalized in RDF[9], and Melita works with its own proprietary ontology language. Additionally, Melita only supports annotation with ontological concepts, whereas MnM only supports annotation with ontological relations, therefore we were forced to redesign the same ontology for use with each tool.

A simple `Visits` ontology was designed for the annotation of the Planet Visits corpus. The ontology contains 11 concepts and 14 relations such as location, date, duration, organization, visitor, visited, host, etc.

For use with the tabular corpus, a very simple `Conference` ontology was designed. The corpus only contains the names of `persons`, `organizations`, `locations` and `dates`, therefore we created a simple ontology containing these four concepts.

Finally, for use with the Baha'i News corpus, the `WordNet`[5] ontology and Kim's own `Kimo`[6] ontology were used.

4.3 Inter-annotation experiment

It was clear that semi-automatic and automatic annotation tools had to be tested against manually tagged corpora and that the manual tagging of a corpus would not be that obvious at all. To gain more insight, we conducted a simple inter-annotation experiment. Two knowledge engineers were asked to annotate all named entities of a subset of 1433 words of the Planet Visits corpus with the `Visits` ontology. The annotations of both experts were compared with the help of the kappa-statistic [10, 11] which corrects for chance agreement. Two important results can be extracted from this experiment. An agreement-rate of 0.79 was obtained, when only considering entities annotated by both experts, showing that the classification with ontological concepts was quite well defined. But when also considering terms annotated by only one of the experts (and left blank by the other), we obtained a much lower rate of 0.435. This leads us to conclude that the task of named entity recognition was quite loosely defined. We also believe that sometimes entities may have been ignored due to lack of relevance. We borrow the notion of *relevance* [12] to describe this phenomenon. For example, it may be asked how relevant it is to annotate distinct references to the same named entity. For example, if the name `Paul Cohen` appears several times within the same paragraph or sentence, does it have to be annotated each time? To give yet another example, consider the annotation of a date such as `11/08/2003`. Is it relevant to say that 11 is a number, or does it suffice to annotate it as a day? Our experience shows that the answer to this question is

[5] http://wordnet.princeton.edu/perl/webwn

[6] http://www.ontotext.com/kim/kimo.rdfs

apparently a complex one, mostly depending on the purpose of the annotation task and the structure of the ontology. Our main conclusion here is that unless the notion of relevance is clearly defined for the annotation task, the agreement rate can be expected to degrade.

5 Results

We start this section by presenting a quantitative evaluation of semi-automatic and automatic annotation procedures, according to the traditional Precision, Recall and F1 measures [12]. Even though these measures are incorporated into the *precision*, *recall* and *reliability* criteria of our evaluation framework, we present them separately due to their important role in the past and for the sake of facilitating comparison with previous evaluations in this field. Please refer to table 2 in section 3 for the definition of these criteria. We then summarize the full scope of our results, according to the four dimensions of the proposed evaluation framework and highlight some of the most important technical drawbacks and features of the evaluated tools. At the end we present a summary of features which may guide the newcomer in his choice.

5.1 The famous Precision, Recall and F1 Measures

Much care must be given to the definition of evaluation set-ups, in order to produce reproducible and meaningful results. Table 4 defines the different set-ups which were used with each tool. Please refer to sections 4.1 and 4.2 for the description of the choice of corpora and ontologies.

Corpora / Tools	Melita 2.0	MnM 2.1	Kim 1.05	C-Pankow
Planet Visits	Visits	Visits	Kimo	Wordnet
Baha'i News	-	-	Kimo	Kimo
Tabular	Conference	Conference	Conference	Conference

Table 4. Different evaluation set-ups. For each tool and corpus we indicate the ontology which was used to annotate the corpus.

The first corpus to be tested was the Planet Visits corpus. To our disappointment, semi-automatic tools, Melita and MnM, failed entirely in learning to replicate annotations on this corpus. Even though much larger corpora could have produced some results, we concluded that the tools were generally not adequate for *unstructured* corpora (typical websites). Based on the results of the previous experiments, we attempted a much simpler corpus: the tabular corpus. Annotation of this corpus produced successful results. In order to have equal bases of comparison among all tools, we also tested Kim and C-Pankow on this corpus with the same ontology. Results are presented in table 5. C-Pankow is excluded from this table since it only produced one annotation on this corpus.

Table 4 further indicates that automatic tools Kim and C-Pankow were successfully evaluated on both *unstructured* news corpora. For Kim we obtained an

Corpora / Tools	Melita 2.0	MnM 2.1	Kim 1.05
Precision	87.8	93.9	100
Recall	90.0	77.5	82.5
F1	88.9	84.9	90.4

Table 5. The traditional Precision, Recall and F1 measures given in percentages for the tabular corpus annotated with the conference ontology

F1 score of 59% (P=63.8%, R=54.8%) on the Planet Visits corpus and a score of 64.9% (P=70.6%, R=60%) on the Baha'i corpus, whereas C-Pankow scored 16.4% (P=21.2%, R=13.4%) and 32.9% (P=36.4%, R=30%) on those corpora.

5.2 The new evaluation framework

According to the proposed evaluation framework, we distinguish among two annotation-specific dimensions: *procedure* and *metadata*. Even though the traditional measures discussed in the previous section are an integral part of procedure criteria, we suggest that a holistic view includes other important criteria as well. In conformance with the procedure criteria presented earlier (Table 2 of Sect. 3), summarized results of the procedure dimension as well as all other dimensions will be presented at the end of this section (Fig. 1). We can see that both semi-automatic tools produce similar results, since they are based on the same information extraction engine. With regard to automatic tools, scores in Figure 1 are based on the annotation of the Baha'i corpus. When comparing the results of Kim with those of semi-automatic tools, the future of the latter tools is clearly under question taking into account the simplicity of the former. Although C-Pankow's results fall slightly behind, it promises to work well on domain-specific corpora [6] where Kim's upper-ontology is of little use.

In the arena of metadata, Ontomat clearly scores best. Interestingly, semi-automatic tools Melita and MnM obtain the same medium scores, whereas automatic tools Kim and C-Pankow fall slightly behind. The equal scores of the semi-automatic tools reflect the fact that they have similar metadata components and that they both author annotations in a similar mark-up based on XML [13]. Even so, both have an interoperability of 0: the XML language lacks formal semantics, therefore the semantics of the annotations are those intended by the authors of the tools. Since both tools follow different naming conventions for the creation of the XML tags (annotations), there is no formal way of establishing that the annotations of one tool are the same as the annotations of the other. For example, considering an ontology which has the concept `House` as a subclass of `Thing`, whenever an entity is annotated as a house, Melita annotates the entity with the tag `<house>`, whereas MnM creates the tag `<thouse>`. The tools follow different naming conventions and due to the lack of semantics of the XML language, there is no way of establishing that both tags refer to the same concept.

We point out that Ontomat's annotations are of high quality, formalized as external annotations in OWL [14]. Additionally, they may be embedded into web pages, simulating internal annotations. Although not an essential requirement of an annotation tool, Ontomat's ontology creation component is also extremely handy. With regard to automatic annotation tools, their low scores can be justified by the fact that they are simple interfaces to remote annotation services, and as such, they don't have a component for editing annotations or ontologies. Low scores in this dimension are mostly due to features that could be present in the tools but are not present in their current state. While such scores might be discouraging to newcomers, experts and researchers may interpret them as new opportunities for development.

Kim distinguishes itself as the tool which scores best in the *scope* criterion. This is due to the fact that it can mark-up all occurrences of the same entity, as belonging to the same instance of the ontology.

Likewise, tools were evaluated according to the domain-independent dimensions (Fig. 1). The final evaluation scores of each dimension are given by the arithmetic mean of the normalized scores of its criteria. We recall that the scores have no absolute meaning, but rather a relative one. Therefore it is only meaningful to compare tools with similar functionality. Concerning full-featured manual tools, the ideal tool should aim to combine the simplicity of Melita's interface, with the annotation-related features of Ontomat. Interestingly all tools scored medium with respect to our general criteria. Melita has a very good documentation, but scores low in scalability. On the contrary, Ontomat is scalable but has very little documentation (only a simple tutorial is available). Also, it has several stability issues, which is not surprising since the tool is officially marked as being in alpha development stage. MnM, on the other hand, scores medium in all general criteria.

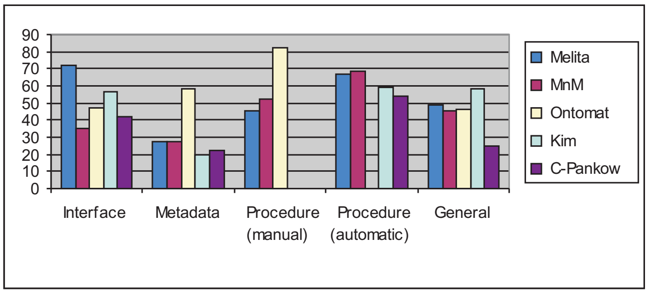

Fig. 1. Comparison of annotation tools.

Concerning the speed of annotation procedures, it was not possible to apply our speed criterion due to the fact that we had no way of finding out which named entities each tool analyzed for annotation. Therefore we have no real indicator for the speed of the tools. Even so, we can affirm that semi-automatic procedures only took several seconds of processing. Kim usually took 4 seconds per page (of approximately 150 words), whereas C-Pankow took several minutes for the same page.

Finally, a brief reference for newcomers is presented in Table 6. Due to lack of space, only some of the most important aspects of our framework are included. With regard to a feature, a classification of `++` indicates *excellence*, a single `+` indicates the feature *exists and is fine*, `+-` stands for *medium to poor* and `-` stands for *absent*. In case a feature is not applicable to a given tool, this is indicated with `na`.

6 Conclusion and Future Work

In this paper we present an evaluation framework for ontological annotation tools. A set of 5 tools were evaluated using this framework. One of the main problems faced by all annotation tools is lack of interoperability. Given the fact that probably one tool is not enough to perform perfect annotation of the web and that probably one should aim at a combined multi-tool approach this is a severe problem. Regarding manual annotation tools our main conclusion is that they do not provide a good enough compromise: some are too focused on the interface, while others focus too much on the metadata. They lack a good compromise on both issues. Regarding semi-automatic tools the main conclusion is they only work with structured corpora, which is not the most common case in the current web. Finally, regarding automatic annotation tools current results are rather impressive, but it is difficult to assess how they could scale. For instance it is difficult to foresee how Kim would perform in domain specific webpages. Therefore, to our understanding, they still have a long way to go before they can be fully used to annotate the current web. The long term goal of the work reported in this paper is to contribute to the automatic annotation field. Our plans include trying to improve one of current tools.

`Criteria / Tools`	Melita 2.0	MnM 2.1	Ontomat 0.8	Kim 1.05	C-Pankow
Documentation	++	+	+-	+	-
Simplicity of the tool	+	+-	-	++	++
Support for Internal Annotations	+	+	+	+	+
Support for External Annotations	-	-	+	-	+
Ontology Editor	-	-	+	-	-
Load multiple ontologies	-	-	-	-	-
Annotate with multiple concepts	+	+	+	-	-
Constraint checking	-	-	+	na	na
Interoperability	-	-	-	-	-
Annotation with concepts	+	-	+	+	+
Annotation with relations	-	+	-	-	-
Ontology Population	-	+	+	+	+
Manual Annotation	+	+	+	-	-
Semi-Automatic Annotation	+	+	-	-	-
Automatic Annotation	-	-	-	+	+

Table 6. Brief summary of features which may be interesting to newcomers.

Acknowledgments Our special thanks go to Bruno Grilo and Rudi Araujo (inter-annotation experiment) and to Claudio Gil and Vitor Oliveira (expert usability analysis). We extend our gratitude to Philipp Cimiano for his constant availability and help with C-Pankow. We would also like to thank Aditya Kalyanpur, Borislav Popov, Enrico Motta, Fabio Ciravegna, Jeff Heflin and Siegfried Handschuh for their replies.

References

1. A. Gómez-Pérez, M. Fernandéz-López, and O. Corcho, *Ontological Engineering.*, Springer-Verlag London Limited, 2004.
2. B. Popov, et al, Kim - semantic annotation platform., in *ISWC*, pp. 834–849, 2003.
3. F. Ciravegna, et al, User-system cooperation in document annotation based on information extraction., in *EKAW*, pp. 122–137, 2002.
4. M. Vargas-Vera, et al, Mnm: Ontology driven semi-automatic and automatic support for semantic markup., in *EKAW*, pp. 379–391, 2002.
5. S. Handschuh, S. Staab, and F. Ciravegna, S-cream - semi-automatic creation of metadata., in *EKAW*, pp. 358–372, 2002.
6. P. Cimiano, G. Ladwig, and S. Staab, Gimme' the context: context-driven automatic semantic annotation with c-pankow., in *WWW*, pp. 332–341, 2005.
7. F. Ciravegna, et al, Amilcare: adaptive information extraction for document annotation., in *SIGIR*, pp. 367–368, 2002.
8. J. Nielsen, Finding usability problems through heuristic evaluation., in *CHI*, pp. 373–380, 1992.
9. Rdf specification, `http://www.w3.org/TR/RDF/`.
10. J. Cohen, A coecient of agreement for nominal scales. *Educational and Psychological measurements*, volume 20, no. 1, pp. 37–46, 1960.
11. J. Carletta, Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, volume 22, no. 2, pp. 249–254, 1996.
12. C. J. van Rijsbergen, *Information Retrieval.*, Butterworth, 1979.
13. Xml specification, `http://www.w3.org/XML/`.
14. Owl specification, `http://www.w3.org/TR/owl-features/`.