# Towards Reasoning and Explanations for Social Tagging

François Bry and Jakub Kotowski

Institute for Informatics, University of Munich
Oettingenstr. 67, 80538 München, Germany
http://pms.ifi.lmu.de

**Abstract.** This paper describes a project aiming at enhancing social tagging with reasoning and explanations. So as to keep with the ease of use characteristic of social media, simple explanations are required. A working hypothesis of the work reported in this paper is that simple explanations require simple reasoning. The approach to reasoning presented in this paper is minimalist: First, it precludes involved forms of reasoning such as refutation or excluded middle; second, it does not need (structural) induction. It is furthermore pragmatic: it incorporates negation as failure and an ad hoc and primitive form of paraconsistent reasoning. Because reasoning is kept simple, a simple and intuitive approach to explanation based only on proof trees is possible. This paper outlines the approach to both reasoning and explanations. Finally, it discusses more sophisticated explanation concepts, based on a notion of proof factorization, that are deemed necessary in the application context considered.

## 1 Introduction

Social tagging, as known from social media such as Flickr (http://flickr.com/), del.icio.us (http://del.icio.us/) and in sciences CiteUlike (http://citeulike.org/) and PennTags (http://tags.library.upenn.edu/), is very promising for professional use for example in education, software development, project management, and collaborative work in sciences. Indeed, social tagging is a means to share semantic annotations within a community [1]. Sharing annotations has been, long before social tagging appeared, widespread within professional, especially scientific, communities.

The professional use of social tagging can be considerably enhanced by reasoning. Indeed, reasoning makes it possible to semantically relate tags to each other in an application dependent manner. A rule might for example express that a tag "bug report" without corresponding tag "processed" induces an implicit "todo" tag. Such rules have to be specified, like the tags themselves, by the user communities. Depending on the application considered, all users could author rules, or only some users particularly experienced or having particular roles.

The research reported about here takes place within a research project KIWI, a code name for "Knowledge in a Wiki", see http://kiwi-project.eu/. The overall objective of KIWI is to design, conceive, deploy and test on a few selected use cases a so called semantic wiki [2], that is a wiki offering user-driven and automatic social tagging, reasoning with content and tags, personalization, version management and explanations. The KIWI wiki will build upon the semantic wiki IkeWiki [3] developed at Salzburg Research.

The use cases investigated in the KIWI project are from software development and project management. A further use case of concern is collecting scientific literature.

We isolated the following issues as challenges of reasoning for social tagging:

**Challenge 1.** To design a language for expressing structured rules and rules on tags, cf. Figure 1, which is intuitive and simple enough for being widely accepted.

The need for structured tags and rules (implying implicit tags) can be demonstrated on the Artigo social tagging system (http://artemis.lmu.de/artigo) designed by Dr. H. Kohle, Art history at University of Munich. Each tag freely chosen by the visitor is assigned two attributes: the user identifier and ranking expressing whether it is the first, second, etc. tag entered for this painting by the user. The rule, the validation of which has been confirmed in practice, is as follows: if the first two tags assigned by at least 95% of the visitors are colours then it is abstract art.

**Challenge 2.** To propose an intuitive form of reasoning well suited to the particular context of social tagging.

**Challenge 3.** To reason with data, that is tags and rules, that in many respect can be seen as fuzzy[1] or incompletely specified, or more formally, in a preaxiomatized shape.

**Challenge 4.** To provide easy-to-grasp explanations for the deductions performed by the systems.

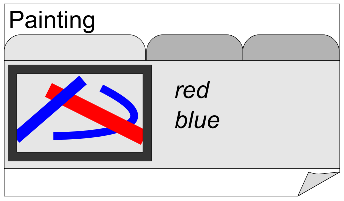

**Fig. 1.** Paintings tagged with colours are likely to be abstract art.

Clearly, the four challenges are closely related to each other. In particular, challenge 4 on explanations induces strong requirements on the design of the

---

[1] Fuzzy is meant in common sense, not referring to Zadeh's fuzzy logic.

rule language (challenge 1), the form of reasoning (challenge 2) and the kind of "fuzziness" of data allowed (challenge 3).

Challenge 1 clearly depends on challenge 2. Indeed, no syntax can be intuitive and simple for being easily understood by a wide user community if it must convey involved forms of reasoning.

In the following is briefly sketched how these four challenges are tackled. Because of the dependencies between the challenges mentioned above, the focus is on challenges 2, 3 and 4. The working hypothesis in this research is that a conveniently simple language syntax should result from the simplicity of the language semantics (understood as reasoning paradigm).

This paper is consciously limited to sketches. A detailed treatment will be provided later. The authors understand the workshop as convenient a place for presenting on-going work, and receiving feedback from other explanation contexts and stressing open issues as far as explanations are concerned.

## 2 Requirements on Reasoning

In a pre-investigation, we recognized the following requirements for the form reasoning to be used in deriving implicit tags from explicits ones:

*Flat and Structured Tags.* Needed are both flat tags, amounting to propositional logic atoms and structured tags amounting to first order predicate logic atoms. Structured tags are desirable for example for *"hotelStars(3), bugReport(14.4.2008)"*, *"locatedIn(downtown)* or *memberIn(Auberges-et-Châteaux)"*, cf. Figure 2, and more generally RDF triples and the like. Note that for example *"locatedIn(downtown)"* can be seen as an RDF triple where the page is an implicit subject.

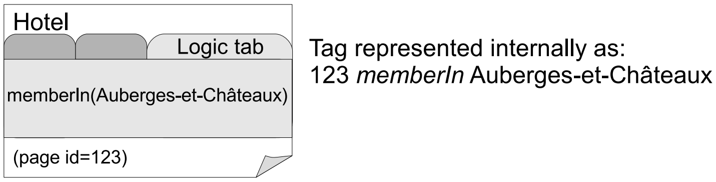

**Fig. 2.** Wiki page with structured tag *"memberIn(Auberges-et-Châteaux)"* internally represented as *"memberIn'(123,Auberges-et-Châteaux)"* (or as an RDF triple *"123 memberIn Auberges-et-Châteaux"*).

*Categoricity.* The reasoning should be categorical in the sense that for each possible information (formula) $F$, either $F$ or $\neg F$ should be provable. The rational for this requirement is to exclude "disjunctive tags". Even though disjunctive tags undoubtedly might be useful for some applications, they basically convey an "alternative worlds" semantics which surely is not the most widely assumed when annotating. Furthermore, the restriction to categorical reasoning simplifies explanations.

*Intuitionistic Reasoning Restrictions.* Requiring witnesses of existence in proving existential statements, that is $\exists x F$ can only be proved if a value for $x$ is shown for which $F$ holds, and to necessarily prove one of $F_1$ or $F_2$ in proving $F_1 \vee F_2$, intuitionistic reasoning is rather intuitive and simple. It excludes both refutation proofs and some forms of reasoning by cases that, arguably, are not familiar to everyone and therefore might be difficult to convey in proof explanations.

It is worth stressing that the choice made here of a sort of intuitionistic reasoning neither is philosophically motivated, nor it is a rejection of standard mathematics. It is no more than a pragmatic choice aiming at simplifying reasoning and therefore explanations. Indeed, the finitist assumption, see next, would make it possible to use reasoning by refutation and cases within an intuitionistic framework.

*Two-Valued Logic.* So as to well fit with a widespread common sense, a two-valued logic is desirable. As a consequence, fuzzy reasoning is not directly expressible. We shall however see under the requirement "Partial Knowledge Reasoning" that there might be a graceful way to achieve some of the usual practical effects of fuzzy reasoning. Note also that the two-valued logic requirement precludes forms of paraconsistent reasoning building upon specific truth values for conveying inconsistency.

*Finitism.* Finite numbers of both individuals and derivable minimal statements can be, if not must be, assumed. Indeed, many applications of social tagging hardly will require deductions yielding infinitely many implicit tags. Whether applications with possibly an infinite tag generation might be meaningful is not further addressed here. It suffices that applications limited to finitely many tags are likely to be frequent.

*Paraconsistency.* Tagging is, in most cases, inherently never perfect and never completed. Even the most sophisticated classification schemes are approximations that rarely are above every criticism. Thus tagging is one of those knowledge representation activities where reasoning is needed before a well-thought, clean specification is available. Furthermore, the collaborative dimension of social tagging adds a cause of "fuzziness": what clearly is, say, a "priority" for me might well not be for one of my co-workers. Thus, a meaningful form of reasoning with inconsistent specifications, that is paraconsistent reasoning, is needed. Following a motive already mentioned several times, a *simple* form of paraconsistent reasoning is needed.

*Negation as Failure.* It is a common understanding of tagging that the absence of a tag means a negation: a book not tagged "computer science" at a university library can safely be taken for a non-computer science book. Thus, negation as failure is the negation of choice, if not the only thinkable one, for reasoning with tags. For the sake of simplicity, no other form of negation is considered in this study.

Note that negation as failure is a graceful and rather intuitive way to ensure categoricity. Note further that negation as failure is relatively easy to explain. Arguably, it is easier to explain than classical logic negation.

*Elementary Arithmetics.* Elementary arithmetics is needed for summing up, computing averages, comparing numbers etc. Such operations are commonly used in classifying things: A "large class" is for example a class with more than a given number of students, etc. Aggregation is needed for computing for example averages.

*Partial Knowledge Reasoning.* So as to appropriately convey the social dimension of social tagging, it should be possible to limit reasoning to those tags set or endorsed by a person or a group and also to compare the outcome of such user- or group-based forms of reasoning.

This requirement can easily be achieved by labeling, that is tagging [sic], the tags by their authors, their endorsers or refuters. Therefore, this requirement is not further discussed in the following.

Note that this approach trivially gives rise to a majority, or "x% supported" deduction: a conclusion can be seen as valid under "majority deduction" (or "x% deduction") if every tag and rule it involves are endorsed by the majority (or x%) of the tag endorsers. A more involved, and surely more satisfying, definition requires that all data involved in a majority (x%) deduction be endorsed by the same majority (x%) of endorsers. This (easy) way, fuzzy reasoning can be approximated in a manner both rather intuitive for the users, and algorithmically rather simple.

*Decidability.* Note finally that a decidable reasoning system is desirable and that this requirement is implied by the finitist assumption mentioned above.

Let us conclude this section by stressing once again that less stringent requirements to the form of reasoning would make much sense. Both for the sake of simplicity and as a first step, a possibly extremist position is adopted in this study.

## 3 Knowledge Representation Scheme

This section describes a knowledge representation scheme which provides ground for a brief discussion of a reasoning scheme and explanations.

*Connectives.* A first-order logic language with the connectives $\wedge$ and $\rightarrow$ is assumed. The language is further assumed to have two special formulas, $\top$ (read verum) and $\perp$ (read falsum) that can be seen as 0-ary connectives. $\top$ and $\perp$ always have the following truth values: $\top$ has the truth value "true" and $\perp$ the truth value "false". The disjunctive connective can for a convenient expression of queries (a concept introduced below) be defined as follows: $F_1 \vee F_2 := \neg(\neg F_1 \wedge \neg F_2)$

*Quantifiers.* The usual two quantifiers $\forall$ and $\exists$ are considered.

Note that "counting existential quantifiers" such as "there exists only one", "there exists at least 2", "there exists at most 5" and "there exists between 3 to 5", that can be expressed using the standard two quantifiers and the usual connectives in general are very useful in practice, for they give rise to much more natural expressions of queries. For the sake of simplicity and without loss of generality, they are not explicitly mentioned in the following.

*General Formulas.* Formulas and subformulas are defined as usual. The polarity of a subformula of a formula $F$ is defined as usual: each occurrence of $a$ has for example positive polarity in $(a \vee \neg\neg a)$ while $b$ for example has negative polarity in $\neg(a \wedge \neg\neg b)$.

$\top$ and $\perp$ are formulas. In order to simplify the phrasing in the following, $\top$ and $\perp$ are no atoms (atomic formulas). They could be considered atoms at the cost of lengthening a bit a few definitions below.

*Formulas Considered in the Reasoning Scheme Proposed* Formulas of three kinds are considered in the following: constructive rules and denial rules (or constraints) using which the application can be specified and queries, using which the explicit and derivable data can be retrieved. These three kinds of formulas are defined as follows.

*Queries.* A query is any (well-formed) formula built up using the connectives $\wedge, \vee, \neg, \perp$, and $\top$, the quantifiers $\forall$ and $\exists$.

The query $\top$, which is not very interesting in itself, is useful for simplifying the definition of constructive and denial rules. The query $\perp$ is more interesting for detecting inconsistencies.

*Constructive Rules.* A constructive rule is an expression of the form $\forall^* Q \rightarrow A$ where $\forall^*$ denotes as usual the universal closure of the formula in its scope, $Q$ is a query (as formerly defined) and $A$ is an atom (or atomic formula). (Note that $A$ cannot be either of $\perp$ or $\top$).

*Denial Rules or Constraints.* A denial rule or constraint is an expression of the form $\neg\forall^*\neg Q$, that is $\forall^*(Q \rightarrow \perp)$, where $Q$ is a query (as formerly defined). Where $\forall^*$ denotes as usual the universal closure of the formula in its scope, $Q$ is a query (as formerly defined) and $A$ is an atom (or atomic formula).

*Range-Restriction.* It is often beneficial in practice to restrict (constructive and denial) rules to range restricted rules. The concept is defined as follows.

Range restriction means that every variable occurring in $A$ or in a subformula $N$ of $Q$ such that $N$ has negative polarity in $Q$ also occurs in a subformula $P$ of $Q$ such that $P$ has positive polarity in $Q$.

*Knowledge Representation with Constructive and Denial Rules.* With range-restricted constructive rules only, one can build up a categorical theory amounting to a set of positive variable-free atoms, that is up to the syntax a relational database.

With constructive rules that are not range-restricted, more general statements can be made, namely universally quantified statements. Knowledge representation with constructive rules amount to a set of formulas of the form $\forall^* A$ where $A$ is an atom.

Denial rules make it possible to constrain, or impose requirements, on the knowledge expressed by the constructive rules, possibly resulting in inconsistencies. Denial rules thus amount to integrity constraints in a relational database.

With social media, especially with wikis and social tagging tools, as well as in many other knowledge representation contexts, inconsistencies are not undesirable at all. Indeed, as already mentioned, social media rely on knowledge representation and reasoning is needed already before a well-thought, clean specification is built up. With "specifications in work", inconsistencies are the witnesses of imperfections to be discarded at later stages. Imperfections (witnessed by inconsistencies) that are of special interest in social media in general and social tagging in particular are those resulting for contradictory tags.

*Tagging.* Under the knowledge representation scheme defined above, that is building upon constructive rules, denial rules and queries, tagging can be formally redefined as specifying constructive rules or denial rules.

This, however, does not specify how a tag, that is a (constructive or denial) rule is attached to a wiki page. This can technically and conceptually easily be realized as follows by syntactically deriving a new language from the tagging language:

- The set of constants of the (tagging) language is extended with the set of the (potential or actual) unique identifiers of the wiki pages.
- Each $n$-ary atom $p(a_1, \ldots, a_n)$ tagging a wiki page with unique identifier $id$ is mapped to the $n+1$-ary atom $p'(id, a_1, \ldots, a_n)$, cf. Figure 2 on page 3. It is assumed that the initial language, to which $p$ belongs, and the resulting language to which $p'$ belongs, do not have non-logical, that is application defined, symbols in common.

## 4 Reasoning Scheme

This section outlines a basic reasoning scheme that stems from the requirements and definitions described in previous sections. The reasoning is described in terms of proof trees on which then explanations are based.

*Basic Reasoning Scheme.* Basically, modus ponens, conjunction introduction, quantifier specialization, and negation as failure suffices to explain the basic reasoning scheme. In more intuitive terms, the basic reasoning scheme can be explained by forward chaining on all constructive and denial rules.

In practice, backward chaining (and reasoning at query evaluation time) can be preferred to forward chaining (and saturation reasoning at update time).

Note that if only queries in which no negation occurs are considered, then negation as failure is not needed. This case amounts to datalog with bottom being interpreted as an atom, that is without necessarily having the truth value false.

*Paraconsistent Reasoning.* The basic reasoning scheme sketched above is paraconsistent in the sense of inconsistency-tolerant. Indeed, if bottom can be derived (using a denial rule), then it is *not* the case that every formula can be derived, too.

This is quite easily, and almost properly proven by assuming that the (constructive and denial) rules are expressed with the logical connective initially introduced above, that is using only $\wedge, \rightarrow, \bot$, and $\top$. This is possible because we have postulated that the further connective $\vee$ is defined as shorthand notation for some formula in which only $\wedge, \rightarrow, \bot$, and $\top$ might occur. Indeed, there is an inconsistency within the constructive and denial rules if and only if it possible to derive $\bot$ from these rules using the basic reasoning scheme sketched above. However, $\bot$ can never be used in evaluating the antecedent of a rule.

Thus, the basic reasoning scheme sketched above cannot be conveyed by a Tarskian model theory. Indeed, with a Tarskian model theory, inconsistencies amount to the non-existence of model, thus making every formula derivable.

In contrast, with the basic reasoning scheme proposed above, inconsistencies are kept, in a sense, local. Consider two sets $S_1$ and $S_2$ of wiki pages. Assume that $S_1$ and $S_2$ are unrelated (by tags) to each other. In such a case, if the tagging of $S_1$ is inconsistent, then this has no effect on what can be derived from the tagging of $S_2$. This emphasises that the reasoning scheme has no refutation.

Arguably, such a "locality of inconsistency" is highly desirable for social tagging.

## 5 Explanation Scheme

The requirements on reasoning and knowledge representation schemes are fulfilled with proofs having the form of proof trees. Such simple proofs simplify explanation. The bearing on explanation is discussed in this and the following section.

*Proof Trees and Forests of Failed Trees.* If only rules in the antecedent of which no negation occur are considered, then the standard notion of proof tree suffice to convey proofs. A proof tree is a (forward or backward) chaining of rule instances.

If negation might occur in rule antecedents, then forests of "failed trees" are needed for conveying negation as failure.

*Rendering.* Explanations are not only concerned with formal content but also to a considerable extent with the manner how this content is conveyed to users. Thus, diagrammatic presentations of proof trees are surely preferable (see for example [4]) over purely textual ones. Diagrams however, are no answers to the blurring effect large proof in general, and proof trees in particular, might cause.

Thus, the authors consider rendering of proof trees and forests a core issue. To the best of their knowledge, this issue has so far not much been investigated. In the rest of this section, first reflections on the issue are presented. The next section presents open issues and open issues for further research on the explanation scheme for social tagging.

*Un/Foldable Trees.* A rather natural, and immediate, way to avoid users feel blurred by large proof trees and forest is to render them with un/folding points. For example, every node in a tree can give a hook for un/folding the subtree it is the root of. A similar un/folding has been successfully used in the visual query language visXcerpt [5,6,7]. Inspiration could also be found in automated theorem proving area in the Interactive Derivation Viewer (IDV) which hides derivations based on interestingness rating [8].

*Minimal Inconsistent Sets.* Marking each minimal inconsistent sets, that is inclusion minimal sets of construction and denial rules involved in derivations of "bottom", is probably desirable. Indeed, this gives ways to outline how much the current state of knowledge representation is imperfect.

*Further Inconsistency-Related Marking.* Marking in a proof tree all those construction and denial rules involved in an inconsistency probably is desirable as well. Indeed, this gives a rather natural and simple way to convey to users that, in a less imperfect state of the knowledge representation, these proof trees might no longer be possible.

Note that more than one such marking is likely to be needed. Indeed, the notion of inconsistency induced by the reasoning scheme considered is, as already mentioned above, local. As a consequence, it makes sense to distinguish the different sources of inconsistencies and mark the derivations accordingly (i.e. have a way of determining all sources of inconsistencies on which the derivation is based). Consider for example the following tags and rules, cf. Figure 3, (assume that all refer to the same wiki page). There are two minimal sets of tags yielding inconsistencies: $S_1 = \{a, b\}$ and $S_2 = \{a, c\}$. Tag $a$ could be marked with inconsistency tags $i_1$ and $i_2$ which would both be inherited by $f$ through the first rule. Tag $b$ could be marked with only $i_1$ which would be inherited by $g$ through the second rule.

## 6 More on Explanations

*Factorizing Explanations.* The explanation scheme sketched above does not address instances. One might expect that, in practical applications, many proofs

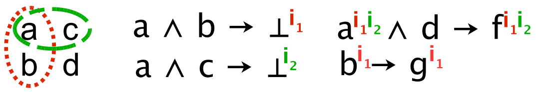

**Fig. 3.** Two minimal sets of tags yielding inconsistencies.

only differ in the instances involved, that is the proof tress would have exactly or almost the same structure, differing only in (individual) constants and variables.

In such a case, it would make much sense to render several proof trees by "factorizing out" the common structure and listing the (individual) constants and variables.

It is worth stressing that such an approach has already been developed and applied with much success for conveying answers to database queries. The celebrated diagrammatic query language QBE, short for Query By Example, delivers answers in such a manner. More recently, "computed answers aggregates" [9] for queries to XML data have been proposed that provide with a much involved instance out-factorization.

However, even though the examples mentioned above show the way, they by no means solve the problem of out-factorizing common parts of proof trees and forests for the explanation scheme considered here.

*Knowledge Model for Explanations.* An additional question is the following: How to express, detect, and use what a user knows so as to simplify, in the sense of shorten, explanations?

This issue can probably be tackled either automatically (cf. [10]) or even better by relying on approaches to adaptation/personalization. Tracking the explanation a user already has seen might be convenient a way to guess what s/he might be assumed to know (see [11] for an example of such an approach).

*Why-not and What-if Explanations.* So far we discussed explanations of fact derivations. A valuable feature, particularly in social media context, would be to also be able to explain why a particular fact was not derived (e.g. why a tag was not applied to a page). Another question we would like to explore is: how to detect and explain what impact would a change of a rule have? These issues could probably be approached by extending and building upon the features already discussed.

*Local Rules.* In the schemes described above (constructive and denial) rules are global. It would be interesting to investigate the possibility of making them local, i.e. only valid in a part of the wiki content.

## 7 Conclusion and Future Work

This paper sketches a possible approach to reasoning and explanation in context of social media and particularly tagging. Because the intended audience of social media is not experts, simple explanations are needed. Simple explanations are obtained by a rather simple knowledge representation and reasoning scheme derived from usage requirements. Issues drawing further investigations are mentioned: proof tree rendering, explanation factorization or explanation of why-not and what-if question.

## Acknowledgements

The research leading to these results is part of the project "KiWi - Knowledge in a Wiki" and has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

## References

1. Bry, F., Wagner, H.: Collaborative categorization on the web: Approach, prototype, and experience report. Technical report, Institute for Informatics, University of Munich, http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2003-12 (2003)
2. Schaffert, S., Baumeister, J., Bry, F., Kiesel, M.: Semantic wikis. To appear in IEEE Software (2008)
3. Schaffert, S.: Ikewiki: A semantic wiki for collaborative knowledge management. 1st International Workshop on Semantic Technologies in Collaborative Applications STICA 06 (2006)
4. Barwise, J.: Logical reasoning with diagrams. Oxford University Press, (ISBN: 0-19-510427-7) (1996)
5. Berger, S., Bry, F., Furche, T.: Xcerpt and visxcerpt: Integrating web querying. Proceedings of Programming Language Technologies for XML (PLAN-X) (2006)
6. Berger, S., Bry, F., Bolzer, O., Furche, T., Schaffert, S., Wieser, C.: Xcerpt and visxcerpt: Twin query languages for the semantic web. Proceedings of 3rd International Semantic Web Conference (ISWC2004) (2004)
7. Berger, S., Bry, F., Schaffert, S., Wieser, C.: Xcerpt and visxcerpt: From pattern-based to visual querying of xml and semistructured data. Proceedings of 29th Intl. Conference on Very Large Data Bases (VLDB03) (2003)
8. Trac, S., Puzis, Y., Sutcliffe, G.: An interactive derivation viewer. Proceedings of User-Interfaces for Theorem Provers, UITP'06 (2006)
9. Meuss, H., Schulz, K.U., Bry, F.: Towards aggregated answers for semistructured data. Proceedings of 8th International Conference on Database Theory (ICDT 2001) (2001)
10. Alexoudi, M., Zinn, C., Bundy, A.: English summaries of mathematical proofs. Proceedings of the IJCAR 2004 Workshop 7 on Computer-Supported Mathematical Theory Development (2004)
11. Fiedler, A.: Natural language proof explanation. Mechanizing Mathematical Reasoning (2005) 342–363