Fuzzy-DL Reasoning over Unknown Fuzzy Degrees

Stasinos Konstantopoulos and Georgios Apostolikas

Institute of Informatics and Telecommunications
NCSR 'Demokritos'
Ag. Paraskevi 153 10, Athens, Greece
{konstant,apostolikas}@iit.demokritos.gr

Abstract. In this paper we describe a fuzzy Description Logic reasoner which implements resolution in order to provide reasoning services for expressive fuzzy DLs. The main innovation of this implementation is the ability to reason over assertions with abstract (unspecified) fuzzy degrees. The answer to queries is, consequently, an algebraic expression involving the (unknown) fuzzy degrees and the degree of the query. We describe the implementation and discuss a use case in the domain of semantic meta-extraction where conventional DL reasoning is not applicable.

1 Introduction

One of the most active areas of research around the construction of the Semantic Web is the bridging of the *semantic gap*, the difference between the automatically extracted, concrete features of web objects and the abstract, semantic features necessary for semantic browsing and querying.

In the domain of multimedia analysis, conceptual modelling technologies are used to bridge this gap, by defining abstract concepts like 'interview' in terms of concrete features, e.g., the recognition of two human figures and speech in a video. Such definitions are made in the context of *ontologies*, representation technologies that capture conceptual knowledge about a domain.

There are two sources of error in this scenario: the video analysis tools used to extract the concrete features (e.g., the appearance of human figures or of a microphone in the video) and the logic rules used to infer abstract features from concrete ones. Since neither of these two levels performs perfectly, erroneous features are going to be assigned at some point; negative feedback from the users of the system is invaluable for improving the system, but it is neither reasonable nor reliable to expect that users accurately identify the source of the error, so that feedback is directed to the party responsible.

In our example, requiring that users giving feedback know about the system-internal definition of the concept 'interview' and are able to tell if the definition is not applicable (e.g., a video showing a conversation between a shopkeeper and a customer) or the recognition was faulty (the video was, in fact, a documentary where a single person describes a statue) is only going to result in sparser and less reliable feedback due to the increased complexity of user input required.

R. Meersman, Z. Tari, P. Herrero et al. (Eds.): OTM 2007 Ws, Part II, LNCS 4806, pp. 1312–1318, 2007.

The problem we are tackling in the work described here is exactly this: given that a user has flagged an abstract feature as wrong, decide whether it is more appropriate to direct this feedback to the video analysis tools or to the ontological definitions. The problem is further convoluted by the fact that it is often desired that such definitions are not absolute, but vague. In our example, consider an ontological rule that states that a video of two people talking *looks like* an interview to a degree of 80%.

This paper is organized as follows: we first provide an overview of vague ontological reasoning, and then proceed to describe a methodology for utilizing negative user feedback in the context of semantic meta-data extraction. Subsequently, we concentrate on our proposed reasoning system, which supports the requirements of this methodology. Finally we draw conclusion and outline future research directions.

2 Reasoning over Vague Knowledge

Ontologies are representation technologies that capture conceptual knowledge about a domain by defining a hierarchy of *concepts*, where more general concepts subsume more specific ones. Concepts are sets of *instances*, or individual objects of the domain. Instances have *properties*, which relate them either to other instances of the domain or to concrete values (e.g. numbers or strings). Properties of the former kind are called *relations* and of the latter *data properties*.

One of the most prominent formalisms for representing ontological knowledge is OWL [1]. OWL is closely coupled to *Description Logics* (DLs) [2], a fragment of first-order predicate logic. DLs give up expressivity in favour of lower computational complexity, but care is taken that their expressivity is sufficient to reason over OWL ontologies [3].

In order to be able to capture 'vague' knowledge—note the 'looks like' in our example above—multi-valued logics replace the binary yes-no valuation of logical formulae with a numerical one, denoting the *degree* to which the formula holds. *Multi-valued DLs* have been successfully used in multimedia information extraction [4] and the ability to model vague concepts has been explicitly stated as a desideratum by the Semantic Web community [5].

Logical formalisms like Description Logics are typically interpreted with *set-theoretic semantics* which define logical connectives and operators in terms of set theory. We shall not here re-iterate these formal foundations, but refer to handbooks of Description Logics [2, Chapter 2]. Informally, unary predicates (concepts) are interpreted as sets of individuals, binary predicates (relations) as sets of pairs of individuals, and the logical connectives as set operations; i.e., concept disjunction is interpreted as set union, concept conjunction as set intersection, and so on.

Binary logics base their interpretations on *crisp* set theory, where an individual's membership in a set gets a binary (true-false) valuation. Multi-valued logics, on the other hand, base their interpretations on vague set-theoretic semantics, where an individual's membership in a set gets associated with one of many (instead of two) possible values, typically real values between 0 and 1.

Fuzzy set theory [6] is such a multi-valued set theory, where the valuation denotes the *degree* to which an individual is a member of the set, or, in other words, the degree to which an individual is a typical member of a set. Fuzzy interpretations are based on algebraic *norms* that provide multi-valued semantics for the logical connectives; the norm that applies to conjunction is called *triangular norm* or *t-norm* and the one for implication is the *implication norm* or *i-norm*. In the work described here we use Łukasziewicz semantics, where the t-norm of the expression $X \wedge Y$ is given by $\max(0, X + Y - 1)$ and the i-norm of $X \rightarrow Y$ by $\max(1, 1 - X + Y)$

The most common approach to implementing multi-valued reasoners is to combine proof algorithms, like *resolution* or *tableaux*, with numerical methods [7,8]. It should noted, however, that all existing reasoning algorithms and implementations require that the degrees of all assertions in the knowledge base are numerical constants, a restriction which renders them unsuitable for our back-propagation methodology described in Section 3 below.

3 Error Back-Propagation

As mentioned in the introduction, we are addressing the issue of analysing erroneous results by a blame assignment system, in order to provide corrective feedback to the level that would be more likely to have introduced the error.

We do this by providing a simple cost measure for the desired changes in the degrees of the concrete features, so that the new system correctly tags the video instance. Given the current system parameters and a new instance of erroneous feature assignment, we need to identify the first-level feature fuzzy degrees that (a) would have yielded 'acceptable' output; and (b) are as close as possible to the degrees calculated by the first-level (video analysis) tools, with respect to Euclidean distance.

User feedback does not give any information for the intermediate level of the system, neither does it include any specific fuzzy degree. Instead it is a binary correct/incorrect opinion, or, at most, a qualitative estimate like 'clearly incorrect', 'almost incorrect', etc. Either way, the system has prior thresholds for translating quantitative membership (fuzzy degrees) to such qualitative descriptions of membership to a concept. In this context, we define as 'acceptable' in point (a) above, a value that satisfies such thresholds for the user's qualitative estimation.

We refer to this scheme as *back-propagation*, since it propagates the error observed at the results of the second level (logic rules) back to the intermediate results of the first level (video analysis). We shall here only briefly outline this method, as it is discussed in detail elsewhere [9]. The goal is to find the concrete feature degree values (first level output) that result in abstract feature degrees (second level output) that satisfy points (a) and (b) above. The method selects a first-level feature and makes its degree an unknown variable, then uses the reasoner to calculate the algebraic relation between this degree and the degree of the abstract feature that was found to be erroneous; this relation has the form of inequalities constraining the possible value assignments. Given the thresholds corresponding to the qualitative description provided by the user, these inequalities are used to calculate thresholds for the concrete feature degrees.

This procedure is repeated for all concrete features, but it should be noted that not all features contribute overall constraints since some proofs might not involve unknown degree values. The goal is now to find the vector of first-level feature degrees that yields an acceptable output and at the same time has the smallest Euclidean distance from the original first-level features. This is achieved by employing an iterative method. An initial acceptable first-level feature vector is found using a coarse search in the feature vector space. Afterwards, the algorithm iterates through the elements in the vector searching for alternate vectors with higher proximity to the original first-level feature vector. The procedure terminates when no more optimization is possible.

4 Reasoning over Unknown Fuzzy Degrees

In order to overcome the limitation of existing fuzzy DL reasoners that all assertions in the knowledge base (KB) are numerical constants, we have designed a novel reasoning methodology which can reason over KBs where some of the degrees as left as variables.

YADLR is a prototype implementation of this methodology, written in Prolog. Its architecture specifies three modules, for all of which multiple implementations are possible. The central module is the *inference engine*, which implements a deduction method like *resolution* or *tableaux*. The inference engine relies on an *algebraic norm* module, which provides semantics for the logical operators. Finally, the *clause representation* module acts as a front-end which translates assertions and queries into YADLR's internal representation, and utilises the inference engine in order to calculate the answer to the query posed.

In its current state of development[1] YADLR implements an SLG Resolution inference engine, the Łukasziewicz set of norms, and a Prolog-term front-end. This last module accepts logical statements in the form of nested Prolog terms, optionally coupled with a fuzzy degree. If the fuzzy degree is omitted it is assumed that its value is unknown and should treated as a variable when calculating derivative fuzzy degrees.

The front end provides reasoning services by converting calls to the service to equivalent logic queries. The three services provided are checking if a given instance is a member of a given concept, retrieving all members of a concept, and calculating all concepts an instance is a member of. All services admit two calling modes, one where the fuzzy degree of the answer to the query is returned, and one which accepts as input a minimum degree and checks whether the query can be answered at a smaller or equal degree of vagueness.

[1] See `http://sourceforge.net/projects/yadlr`

4.1 Fuzzy SLG Resolution

Many common resolution calculi and algorithms are based on the *resolution rule* [10]. The resolution rule specifies the conditions under which a clause R can be deduced from two clauses C_1 and C_2. Resolution is a sound and complete deduction rule for first-order logic, which is to say that it identifies *all* and *only* the cases where two clauses semantically support a third clause.

Selection Linear resolution for General Logic Programming (SLG Resolution, [11]) is a deduction algorithm based on the resolution rule. Given a *query* or *goal* G to prove, SLG resolution makes a left-to-right pass on the literals l_i of G, identifying which (if any) clause in the Knowledge Base (KB) can resolve against l_i. Effectively, each l_i is replaced by the premises of a clause that has l_i as its conclusion. The repeated application of this process takes G through a series of transformations G', G'', etc. until a clause is reached that is either obviously true, i.e., a known logical tautology or a (set of) ground fact(s)—or obviously false, i.e., a logical contradiction.

Literals that, at a given step, can be neither proven nor disproven are placed in a *delay list*. If further down the proof a delayed literal is proven, it is removed from the list; if contradicted the proof fails. A successful proof with an empty delay list means that the formula is satisfied by the background. If, on the other hand, the delay list cannot be closed, a conditional answer is returned which means that the formula is *satisfiable* (subject to the items remaining in the delay list) but not necessarily *satisfied* by this particular KB.

Crisp SLG Resolution is the inference apparatus behind deductive database systems like DATALOG and disjunctive DATALOG. In the domain of Description Logics, the KAON2 system[2] reasons by reducing DL programs to their disjunctive DATALOG equivalents and then using resolution-based reasoning services originally designed for disjunctive DATALOG [12].

In YADLR a fuzzy variation of the SLG algorithm is implemented, where each transformation checks whether the fuzzy degree of the result is above a threshold, and only admits transformation steps that pass this threshold; a successful proof is one that proves that the degree at which the goal is supported by the knowledge base, is above a user-specified threshold. A further refinement of this algorithm, discussed in the following section, also handles unknown fuzzy degrees (in the knowledge base as well as in the goal) and proves algebraic relations between these unknown values.

4.2 Handling Unknown Fuzzy Degrees

When fuzzy values are left as variables in either the knowledge base or the goal G, YADLR effectively restricts the range of fuzzy values that the original clause G admits. More specifically, when ground facts are used in some transformation, the degree of the derivative of G cannot assume certain values if the transformation is to be valid, as specified by the set of norms in use.

[2] See also `http://kaon2.semanticweb.org/`

If all fuzzy degrees are known in advance, this restriction can be immediately checked and the transformation can, accordingly, be accepted of rejected, as shown in the first approximation of the YADLR algorithm in the preceding section. If unknown fuzzy degrees are involved, then each application of the t-norm builds up an ever more restrictive set of inequalities that must be satisfied. At each application of the basic resolution step, the inference module uses the i-norm found in the algebraic norms module, and the latter returns this new set of restrictions.

As seen above, the inference engine defers to the norms module the calculation of the degree of each derivation of G. The i-norm implementation should be able to handle KB assertions where the degree is not specified, but left as a free variable.

In fact, such degrees are not completely unspecified, but possibly restricted by previous iterations, in the same way that the degree of the goal gets restricted. As the proof proceeds, the degree of each node gets calculated using the algebraic norms; whenever assertions with unbound fuzzy degrees are encountered, the admissible values for these variables get restricted within the range that would yield the required valuation for the overall expression, which builds up to a system of linear constraints that is solved using the clp($\mathcal{Q}, \mathcal{R}$) constraint linear programming library [13].

At the end of this process, and if there are any open branches, one collects at the leaves of the open branches a system of inequations. This system specifies the admissible values for the unbound degrees, so that the original formula at the root of the tree is satisfiable.

It should be noted that some transformations might be invalid even when unknown values are involved, as we might, for example, end up requiring that $x < 0.3$ and $x > 0.5$ simultaneously, but in general a lot more (conditional) solutions will be admitted than in a situation where all values are known.

The answer to the logical query is a disjunction of sets of inequalities, involving the unknown fuzzy degrees in the KB and the query.

5 Conclusions and Future Work

In this work we have proposed a novel method for reasoning over fuzzy Description Logics and demonstrated how it can be useful in improving the accuracy of meta-data extraction from multimedia content. More particularly, we have discussed how to reason over knowledge bases that include assertions of an unknown fuzzy degree, and how this is useful to error back-propagation in a fuzzy DL system.

At its current state of development, the system implements the general SLG resolution algorithm with the Łukasziewicz norms for multi-valued semantics. Planned future development includes designing and implementing a resolution methodology that is optimized for reasoning over Description Logic knowledge bases, as well as implementing more of the various algebraic norms proposed.

We are also planning to look for further use cases for our methodology besides error back-propagation for correcting meta-data extraction systems. Such use cases might include decision support systems where not all the parameters of the problem are known, but discovering the relation between the unknown parameters can provide important information.

Acknowledgements

The work described here was partially supported by DELTIO,[3] a project funded by the Greek General Secretariat of Research & Technology. DELTIO focuses on evolving ontologies to analyse multimedia content, with an application to news broadcasts.

References

1. Smith, M.K., Welty, C., McGuinness, D.L.: OWL web ontology language. Technical report, World Wide Web Consortium (February 2004) W3C Candidate Recommendation, http://www.w3.org/TR/owl-guide/
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
3. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics 1(1), 7–26 (2003)
4. Meghini, C., Sebastiani, F., Straccia, U.: A model of multimedia information retrieval. Journal of the ACM 48(5), 909–970 (2001)
5. Kifer, M.: Requirements for an expressive rule language on the semantic web. In: Proc. of W3C Workshop on Rule Languages for Interoperability (2005)
6. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
7. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: The fuzzy description logic f-SHIN. In: Proc. of the International Workshop on Uncertainty Reasoning For the Semantic Web (2005)
8. Straccia, U.: Reasoning within fuzzy Description Logics. Journal of Artificial Intelligence Research 14, 137–166 (2001)
9. Apostolikas, G., Konstantopoulos, S.: Error back-propagation in multi-valued logic systems. In: Proc. International Conf. on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, India, December 13–15 (2007)
10. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the ACM 12(1), 23–41 (1965)
11. Chen, W., Warren, D.S.: Tabled evaluation with delaying for general logic programs. Journal of the ACM 43, 20–74 (1996)
12. Hustadt, U., Motik, B., Sattler, U.: Reasoning for description logics around SHIQ in a resolution framework. Technical Report 3-8-04/04, Forschungszentrum Informatik, Karlsruhe (2004)
13. Holzbaur, C.: OFAI clp(Q,R) manual, edition 1.3.3. Technical Report TR-95-09, Austrian Research Institute for Artificial Intelligence, Vienna (1995)

[3] See also http://www.atc.gr/deltio/