
Automatic extraction of knowledge from student essays

Maria Vargas-Vera* and Emanuela Moreale

Knowledge Media Institute (KMi)

The Open University

Milton Keynes, MK7 6AA. England, UK

m.vargas-vera@open.ac.uk; ema@cs.nott.ac.uk

*Corresponding author

Abstract: This paper presents a characterisation of argumentation in student essays and analyses patterns for extracting knowledge from them. Having analysed their complexity in light of the kinds of logic that may be used in an automatic argumentation extraction system, the main characteristic of these patterns appears to be the polymorphism of the pattern variables. Therefore, systems that learn patterns automatically ought to be able to generate many-sorted logic formulae, so that polymorphic types may be associated to the extraction slots (or – equivalently – to a Logic Formulae). An analysis of existing (pattern learning) systems was carried out to gauge the possibility of using them within our framework. However, we concluded that none of the existing systems can handle our requirements. Finally, we present our vision of an agent-based student portal as the front-end of a system that can locate argumentation links in a student essay and integrates with related educational services.

Keywords: automatic extraction of knowledge, student essays, agent-based student portal.

Reference to this paper should be made as follows: Vargas-Vera, M. and Moreale, E. (xxxx) ‘Automatic extraction of knowledge from student essays’, *Int. J. Knowledge and Learning*, Vol. X, No. Y, pp.000–000.

Biographical notes:

Dr Maria Vargas-Vera obtained her PhD from the Artificial Intelligence Department at Edinburgh University. Since October 2000, she has been working on the high-profile AKT project (Advanced Knowledge Technologies) at the Knowledge Media Institute, The Open University,

Copyright © 200x Inderscience Enterprises Ltd.

England, UK. Her research focuses on the use of Ontologies in Natural Language Processing. In particular, her research is on Ontology-Based Information Extraction, Question Answering, Automated Assessment of Student Essays, Semi-Automatic Construction of Ontologies from text and Ontology mapping using Agents. She is the main author of MnM – a tool for Semi-Automatic Population of Ontologies from Text and AQUA, a Question Answering System over the Web. She is a member of program committees of international conferences/workshops.

Emanuela Moreale holds an M.Sc. from the University of Leicester and is currently a research student at the Learning Sciences Research Institute (LSRI) of the University of Nottingham. Previously a Research Fellow at the Open University, she has worked on various projects relating to software agents as well as automated assessment and argumentation in student essays.

1. Introduction

Full understanding of text has been one of the goals of the Natural Language Processing (NLP) community. This goal is far from being accomplished with current technology. Therefore, a more modest approach to text understanding might be aimed for: a lightweight understanding. This lightweight understanding is performed using shallow parsing and Information Extraction techniques coupled with Machine Learning technology.

The knowledge acquisition phase relies on finding patterns in the text. These patterns are domain specific. As Riloff (1996) pointed out, a manual library of patterns is time consuming to produce and the process tends to be error-prone. In Riloff's view, a good dictionary takes about 1500 person-hours to populate – this estimation was computed by constructing a handcrafted dictionary of patterns in the terrorist domain (Riloff 1996). Therefore, there is a need for the use of Machine Learning technology in the automatic creation of library of patterns. Crystal (Soderland *et al.*, 1995) is an example of such a tool that uses Machine Learning techniques to create a library of patterns.

Machine Learning technology requires a training step for learning patterns of a specific type. The learning is performed showing positive examples of the concept to be learnt. However, it is worth noting that, in some systems, a set of negative examples may be also required. The training in systems

like MnM (Vargas-Vera *et al.*, 2002) is carried out with the assistance of an ontology. In this particular tool, training examples are annotated using the names of the slots of the classes in the ontology. As a second step, MnM requires a learning phase that is performed using Crystal. Therefore, in MnM, an ontology plays two roles:

1. it helps in the extraction of a library of patterns
2. it is used for the reasoning, in particular for generalisation/specialisation (Vargas-Vera *et al.*, 2001).

The former role is carried out during the knowledge acquisition phase, while the latter role takes place during the reasoning phase. MnM has so far been tested in the domain of news articles (Vargas-Vera *et al.* 2002, Vargas-Vera and Celjuska, 2004).

Previous work in extraction of argumentation links used a light weight approach involving cue phrases recognition and simple patterns that were represented as regular expressions. This solution was presented in (Moreale and Vargas-Vera, 2003). The tool implementing this solution, called SVE, allows visualisation of links of argumentation, which are of different types (and visualised in different colours), depending on the type of argumentation. The type of the links is determined each time on the basis of the encountered cue phrases. However, in this paper, we intend to build on this work by describing an approach that proposes to go beyond cue phrases recognition or very simple patterns which can be represented as regular expressions.

Our suggested framework will make use of Information Extraction and Machine Learning technology. In this framework, the knowledge acquisition is performed using an ontology which specifies classes and slots to be filled using the extracted information. But, in addition, we require that polymorphic types can be associated to the slots used for extraction (or the equivalent polymorphic types be associated to a Logic Formula).

We have explored several tools that appeared to suit our purposes, such as RAPIER, BETH and FOIL. As these tools extract relations in a restricted way, we were hoping that it may be possible to adapt these systems to our own purposes. Our main requirement is that it needs to be possible to associate polymorphic types to the slots used for extraction (or the equivalent polymorphic types associated to a Logic Formulae).

Our main contributions are

- To provide a characterization of links of argumentation in student essays.
- To analyse patterns required for finding argumentation links in an educational scenario. In particular, we are interested in patterns which can be extracted from student essays. In our analysis, we have found that these patterns should have polymorphic types associated to a Logic Formulae or polymorphic types associated to the slots of a class. Finally,
- To provide a vision of an agent-based student portal for an e-learning scenario. The student essay argument visualiser would be part of this student portal and one of the services provided through it.

This paper is organized as follows: Section 2 presents a characterisation of types of argumentation in a student essay. Section 3 discusses student essay patterns and introduces many-sorted logic in the context of our domain of interest (student essays). Section 4 presents several systems which were candidates for automatic extraction of arguments in a student essay. Section 5 presents a proposal for a student portal for an e-learning scenario. Finally, Section 6 gives our conclusions and future work.

2. Characterization of links of argumentation in a student essay

Our contribution in the area of argumentation links consisted in depicting a characterization of links of argumentation in student essays. Our approach was to first research what argumentation taxonomies were already available in the literature with the aim of modifying them to suit our purposes. The closest domain for which argumentation taxonomies existed was that of academic research papers. The next few sections will summarise our work in this area, but full details on the research background and the process of devising such argumentation taxonomy can be found in (Moreale and Vargas-Vera, 2003).

Because research papers are characterised by pretty well-established structural conventions, an important strand of research has focused on this aspect, producing metadiscourse taxonomies such as Swales's CARS model. Swales (1990) synthesised his findings that papers present three moves: authors first establish a territory, then a niche and finally they occupy this niche. Although his analysis targeted only the introductory part of an academic research paper, his model has nevertheless been influential. In fact, Teufel (Teufel *et al.*, 1999) extended Swales's CARS model by adding new moves to cover the other sections in a research paper. Given their focus on automatic summarisation, their annotation

schema aimed to mark the main element in a research paper: its purpose in relation to past literature. They classify sentences into background, other, own, aim, textual, contrast and basic categories (see Table 1).

Type	Description
BACKGROUND	Statements describing some (generally-accepted) background knowledge
OTHER	Sentences presenting ideas attributed to some other specific piece of research outside the given paper
OWN	Statements presenting the author's own new contributions;
AIM	Sentences describing the main research goal of the paper;
TEXTUAL	Statements about the textual section structure of the paper;
CONTRAST	Sentences contrasting own work to other work
BASIS	Statements to the effect that current work is based on some other work or uses some other work as its starting point

Table 1 - Teufel's Annotation Scheme (slightly modified)

Another relevant taxonomy is Hyland's metadiscourse schema (Hyland, 1998), which distinguishes between textual and interpersonal types in academic texts (see Table 2). Textual metadiscourse refers to devices allowing the recovery of the writer's intention by explicitly establishing preferred interpretations; they also help form a convincing and coherent text by relating individual propositions to each other and to other texts. Interpersonal metadiscourse alerts readers to the author's perspective towards both the information and the readers themselves: it therefore expresses a writer's persona.

Category	Function	Examples
Textual Metadiscourse		
Logical connectives	express semantic relation between main clauses	In addition, but, therefore, thus, and
Frame markers	explicitly refer to discourse acts/text stages	Finally, to repeat, our aim here, we try
Endophoric markers	refer to information in other parts of the text	Noted above, see Fig 1, table 2, below
Evidentials	refer to source of information from other texts	According to X / Y, 1990 / Z states
Code glosses	help reader grasp meanings of ideational material	Namely / eg / in other words / such as
Interpersonal Metadiscourse		
Hedges	Withhold writer's full commitment to statements	Might, perhaps, it is possible, about
Emphatics	Emphasise force of writer's certainty in message	In fact, definitely, it is clear, obvious
Attitude markers	Express writer's attitude to propositional content	Surprisingly, I agree, X claims
Relational markers	Explicitly refer to/build relationship with reader	Frankly, note that, you can see see
Person markers	Explicitly reference to author(s)	I, we, my, mine, our

Table 2 - Hyland's Taxonomy: Functions of Metadiscourse in Academic Texts

We were also influenced by generic ontologies for scholarly discourse and in particular ScholOnto, a project aiming to model arguments in academic papers and to devise an ontology for scholarly discourse Buckingham Shum *et al.*, 2002). As part of their project, they have produced ClaiMaker, a tool to manually enter (also visualise and search for) claims found in research papers. ClaiMaker claims are classified as general, problem-related, taxonomic, similarity or causal (Table 3). While such classification seemed generic enough, it should be noted that ClaiMaker is mainly meant for academic papers: it sees an academic research paper as a set of inter-linked parts; also, statements in one paper are manually linked with statements in others, leading to a network of cross-referring claims being gradually constructed. Our motivation, however, was different, because we were dealing with (mainly individual) student essays. As a result, our argumentation classification also needed to be somewhat different.

Link Type	Link
<p>General various useful links</p>	Is about, uses / applies / is enabled by, improves on, impairs, other link
<p>Problem-related Links to connect to concepts that are research problems</p>	Addresses / Solves
<p>Supports / Challenges Links to use for connecting evidence and arguments to concepts that are hypotheses or positions taken by the author</p>	Proves, refutes, is evidence for, is evidence against, aggress with, disagrees with, is consistent with, is inconsistent with
<p>Similarity Links to tie together similar concepts, or concepts to be specified as different</p>	Is identical to, is similar to, is analogous to, shares issues with, is different to, is the opposite of, has nothing to do with, is not analogous to
<p>Causal Links to tie up causes and effects, or indicate that certain conditions have been eliminated as possible causes</p>	Predicts, envisages, causes, is capable of causing, is prerequisite for, is unlikely to affect, prevents

Table 3 - Rhetorical Relations Used in ClaiMaker

As a first step in our argumentation taxonomy building efforts, we identified categories of possible arguments in a student essay. This categorisation was mainly based on a preliminary manual analysis of essay texts, with some categories derived from ClaiMaker. Some input came from the other categorisations described above. This resulted in too many categories to manage effectively and, after some revision cycles, we ended up with the categorisation presented here. Our argumentation categories are: definition, reporting, positioning, strategy, problem, link, content/expected, connectors and general.

Compared to Teufel's annotation scheme (Teufel *et al.*, 1999), our schema lacks an AIM category, as student essays implicitly aim to answer the essay question. Similarly, Teufel's distinction between OTHER and OWN (troublesome for human annotators) is irrelevant in our domain. Conversely, the content/expected category is student essay-specific: it includes cue phrases identifying content expected to be found in the essay. Further details can be found in (Moreale and Vargas-Vera, 2003).

Table 4 shows our taxonomy of links of argumentation:

Category	Description	Cue phrases (examples)
DEFINITION	Items relating to the definition of a term. Often towards the beginning. IS_ABOUT, COMPARISONS	is about, concerns, refers to, definition; is the same; is similar /analogous to;
REPORTING	Sentences describing other research in neutral way	“X discusses”, “Y suggests”, “Z warns”
POSITIONING	Sentences critiquing other research VIEWPOINTS	“I accept”, “I am unhappy with”, “personally”;
STRATEGY	Explicit statements about the method or the textual section structure of the essay	“I will attempt to”, “in section 2”
PROBLEM	Sentences indicating a gap or inconsistency, question-raising, counter-claiming	“There are difficulties”, “is problematic”, “impossible task”, “limitations”
LINK	Statements indicating how categories of concepts relate to others TAXONOMIC, EVIDENCE, CAUSAL	“subclass of”, “example of”, “would seem to confirm”, “has caused”
CONTENT/ EXPECTED	Any concept that the tutor expects students to mention in their essay. Tutor-editable	Essay-dependent
CONNECTORS	Links between propositions may serve different purposes (topic introduction, support, inference, additive, parallel, summative, contrast, reformulation)	“With regard to”, “As to”, “Therefore”, “In fact”, “In addition”, “Overall”, “However”, “In short”
GENERAL	Generic association links	“is related to”

Table 4 - Our Taxonomy for Argumentation in Student Essays

In our student argumentation viewer system, each category (shown in Table 4) recognised simple patterns formed by cue phrases. However, in the present paper, we are extending our previous work on argumentation links in student essays: the next section will show examples of more advanced patterns. These patterns use variables that are bound dynamically. Then, the meaning of the quantifier is determined by the type of the variable it binds.

3. Student essays patterns

We have gained some experience in the automatic extraction of patterns from text, which could be described as either un-structured or semi-structured text. However, in our previous domains of application (news articles and job postings), the kinds of patterns were restricted to a unary type for each of the variables in the pattern. We only wanted to extract names of people, locations, etc. In other words, each of these slots was associated to a unique type.

3.1 Patterns structure

In this section, we present the patterns encountered in a student essay. These patterns show the requested main properties for a needed learning algorithm which should be embedded in an Information Extraction system.

Some examples of the encountered patterns in our new case study are for example the following:

x is against y; where x is a person and y is a polymorphic type which can be either a person or a technique or theory.

x support y; where x is a person and y is either a person or a technique or theory.

These patterns require an extraction algorithm which learns formulae in a many-sorted logic. We want to allow any variable to be used for any type, but – in a typical programming language – this will require explicit type declarations. In logic, this can be accomplished with bounded quantifiers.

A simple scenario in arithmetic involving real and integers numbers is used to briefly illustrate a many-sorted logic. Let us imagine that we have two separated universes, for example real and integer numbers and we want to define some functions and relations connecting them. These two universes (real and integers numbers) have their own arithmetic. In a programming language, such as C++, we generally overload, for instance, the symbols of equality, comparisons and variables by using the same symbols in both universes. A function which links both universes is for example, $real(x)$ where the function **real** is mapping an integer into its corresponding real value. In a logic language, describing the same situation will require having two sorts of variables (real and integer variables). Then the meaning of the quantifier will be determined by the

type of the variable it binds. Therefore, the same example can be re-written as follows:

$$\forall x (\text{int}(\text{real}(x)) = x)$$

Coming back to our scenario (patterns from student essays), we notice that our universes are ranging from techniques and papers to theories. A required function in our scenario is, for example, $\text{unify}(y,z)$ where the function **unify** is mapping y to the type of z . This function is slightly different from the unification predicate of Prolog (Clocksin and Mellish, 1981). A language for many-sorted logic should provide a many-sorted unification algorithm. This many-sorted unification algorithm should return a set of substitutions instead of a single one, as in the unsorted case. Further details on the many-sorted unification algorithm can be found in (Walther 1985).

Let us consider the pattern “**x supports y**” where x is a person and y denotes either a person or a technique or theory. In this pattern, y could be bound to the name of a person or to that of a technique or theory. Then, the formula looks like:

$$\forall x : \text{person} \exists z : \text{person} \exists y ((\text{unify}(y,z) = y = z) \text{ and } \text{support}(x,y)) \text{ or}$$

$$\forall x : \text{person} \exists z : \text{technique} \exists y ((\text{unify}(y,z) = y = z) \text{ and } \text{support}(x,y)) \text{ or}$$

$$\forall x : \text{person} \exists z : \text{theory} \exists y ((\text{unify}(y,z) = y = z) \text{ and } \text{support}(x,y))$$

where $x : \textit{person}$ means x has sort person.

In conclusion, we can say that the suggested many-sorted logic should provide us with functions that can perform mappings between several universes (such as person, techniques, theories). However, further research needs to be carried out in this direction of learning formulae of a many-sorted logic.

4. Automatic Extraction of Arguments in a student essay

Research into identification of arguments in research papers has relied on a conceptualisation of academic paper structure: a paper is typically seen as containing an introduction, results and other interesting sections (e.g. paper contributions, usually identified through some heuristics). However, student essays present a somewhat different challenge: while containing background and approach comparisons, they do not usually contain

original contributions to knowledge. More importantly, their structure is less predictable than that of academic papers and cannot therefore be totally relied upon in devising a strategy for argument extraction.

Extraction of entities from text has been explored in the Information Extraction community. However, in order to accomplish our main goal (determining the type of argumentation used in students essays), we need to evaluate some relational learning systems which can be obtained off-the-shelf.

Let us start our analysis by revising some relational information extraction systems, as these systems seem well suited to our task. There are several systems for information extraction that use relational learning (i.e. systems which learn relations between entities). Examples of these systems include for instance, RAPIER (Califf, 1998), BETT (Mooney *et al.*, 2004) and FOIL (Quinlan and Cameron-Jones 1993, Mitchell 1997). These systems use Inductive Logic Programming (ILP). We describe these systems in next section.

4.1 Relational systems

It is worth reminding the reader that relational learning is defined in this paper as the task of learning relations between entities (Lloyd 2000). In relational learning, individuals/objects can be represented by terms with more complicated structure than constants. For instance, they can be sets, lists, graphs and so on, as opposed to the attribute-value case, where individuals/objects are represented by constants.

A set of information extraction systems using relational learning is presented below. We experimented with these systems in order to assess the main restrictions of each of them and to evaluate if we could use them for the automatic extraction of frequently occurring patterns in a student essay.

RAPIER was inspired by Inductive logic programming systems. It has ideas from GOLEM (Muggleton and Feng, 1992), CHILLIN (Zelle *et al.*, 1994) and PROGOL (Muggleton, 1995).

The RAPIER system (Califf and Mooney, 1997) learns single-slot extraction patterns. A Rapiere extraction pattern consists of three slots: the Pre-filler slot, the Post-filler slot and the Filler pattern. The first one plays the role of left delimiter, the second is the right delimiter, whilst the third

describes the structure of the extracted information. From our experiences using RAPIER, we can conclude the following points:

1. One of the major problems in using RAPIER was that users have to fill templates for each document in the training set.
2. The user has also to do POS tagging (using the Brill tagger (Brill 1994)) for each document in the training set.
3. If steps 1 and 2 are not correctly performed, then the learning component cannot learn useful extraction rules. Therefore, there is a necessity for the automatisatio n of these processes.

However, the main restriction of RAPIER is that polymorphism is not supported in any way. Therefore, RAPIER is not suitable for extraction of patterns from student essays. We also tested RAPIER in the Jobs adverts domain: job title, salary, experience required, location and so on. In the jobs domain, RAPIER learns patterns which later in the extraction phase obtain high precision (90%).

BETH uses a covering algorithm like that of any ILP algorithm. Experiences with BETH suggest that the system learns a set of restricted theories of Horn clauses (without negation). In general, BETH is a robust piece of software. However, like all ILP systems, it presents the problem of large amounts of data, since the hypothesis space of possible logic programs is extremely large. From the point of view of the user, the usability of BETH needs to be improved. Currently, it requires an experienced linguist to write the background knowledge specification, the domain specification and the sets of positive/negative examples in Prolog notation. In a similar fashion, BETH is not suitable for our purposes, as it does not handle polymorphic types.

FOIL is a system that learns a set of First Order Formulae. The set of rules learnt by FOIL are Horn Clauses with two exceptions. The rules learnt by FOIL are more restricted than Horn Clauses, since it is not allowed to pass predicates as arguments and, secondly, clauses in the body of a clause can be negated. This gives more expressivity to the FOIL rules. Experiences with FOIL suggest that it is robust software which allows learning First Order Logic formulas from relation specifications and a set of examples. In particular, we used FOIL with KMi news articles describing academic events. However, the type of the arguments in a predicate can take only one value, whilst in our framework (e-learning scenario), we would like to include polymorphic types.

At first, it seemed that one of these systems could be used in our particular problem extraction of relations between entities. However, after using

them, we realised that we needed a more sophisticated system, such as a system which can learn many-sorted logic formulae. However, we believe that FOIL could be extended to handle our requirement. Therefore, we introduced a simplified example tested in FOIL. This example extracts the concept *visitor* from a given specification.

The specification given to FOIL is shown below. The readers should note that the type associated to the predicate arguments needs to be defined in the specification. Currently, FOIL only takes one type for each of the arguments in a predicate.

Types:

Person: {David, Tony, Maria, Victoria, Martin, Clara}

Organization: {Open, Sheffield}

Relations:

people(Person)

{people(David), people(Maria), people(Tony),
people(Victoria), people(Martin), people(Clara)}

places(Organization)

{place(Open), place(Sheffield)}

visitor(Person, Organization)

{visitor(David, Open), visitor(Maria, Sheffield),
visitor(Tony, Open), visitor(Victoria, Open),
visitor(Martin, Open), visitor(Clara, Sheffield)}

works(Person, Organization)

{works(David, Sheffield), works(Tony, Sheffield),
works(Victoria, Sheffield), works(Martin, Sheffield),
works(Clara, Open)}

The results obtained by FOIL are the set of First Order Logic formulae:

visitor(A,B) :- ~works(A,B), people(A), place(B).

The **visitor** rule can be interpreted as *A is a visitor of B if A does not work in B and A is a person and B is a place.*

In conclusion, we can say that an automatic system for extracting argumentation links will require a relational learning algorithm which can

extract relations with polymorphic types associated to the arguments. Further research needs to be carried out in this direction.

5. Architecture of an Agent-based Student Portal

This section describes our proposed architecture for a student semantic portal. Architecturally, a semantic portal consists of a user who has access to services, repositories and databases through an interface. Figure 1 gives an overview of the overall architecture in the e-learning scenario and specifies details of services in the e-learning domain.

In this architecture, the first step would be registering each service with a registry (not shown), so that services can then be invoked through the service broker. The broker is a central component in this distributed architecture: it allows communication between service providers and requesters. In particular, it attempts to match a request for a service to the closest service that can provide that functionality. Services interact with resources and, in particular, subscribe to relevant ontologies. Other resources include databases and documents published on the internet.

We envision a scenario where educational services can be mediated on student behalf. The user/student will confirm that suggestions are acceptable. The advantage of having a semantic portal is that students need not look for courses distributed across many locations (unlike current solutions). Moreover, semantic services perform inferences in the background (taking into account student preferences) as opposed to having users manually searching the traditional way.

An e-learning portal might include services such as smart question-answering, exam marking, intelligent tutoring systems, online courses and a service to help students improve their essays. Of these services, we have so far dealt with the implementation of a question-answering service (AQUA) and SVE, a visualization tool for student essays (Moreale and Vargas-Vera, 2003). AQUA searches for answers in different resources such as ontologies and documents on the web. AQUA is described in detail elsewhere and we refer the reader to these papers (Vargas-Vera and Motta, 2004; Vargas-Vera *et al.*, 2003). We envisage the use of these components as part of a student semantic portal, seen as a door to obtaining knowledge which may be mediated by a set of semantic services (Moreale and Vargas-Vera, 2004).

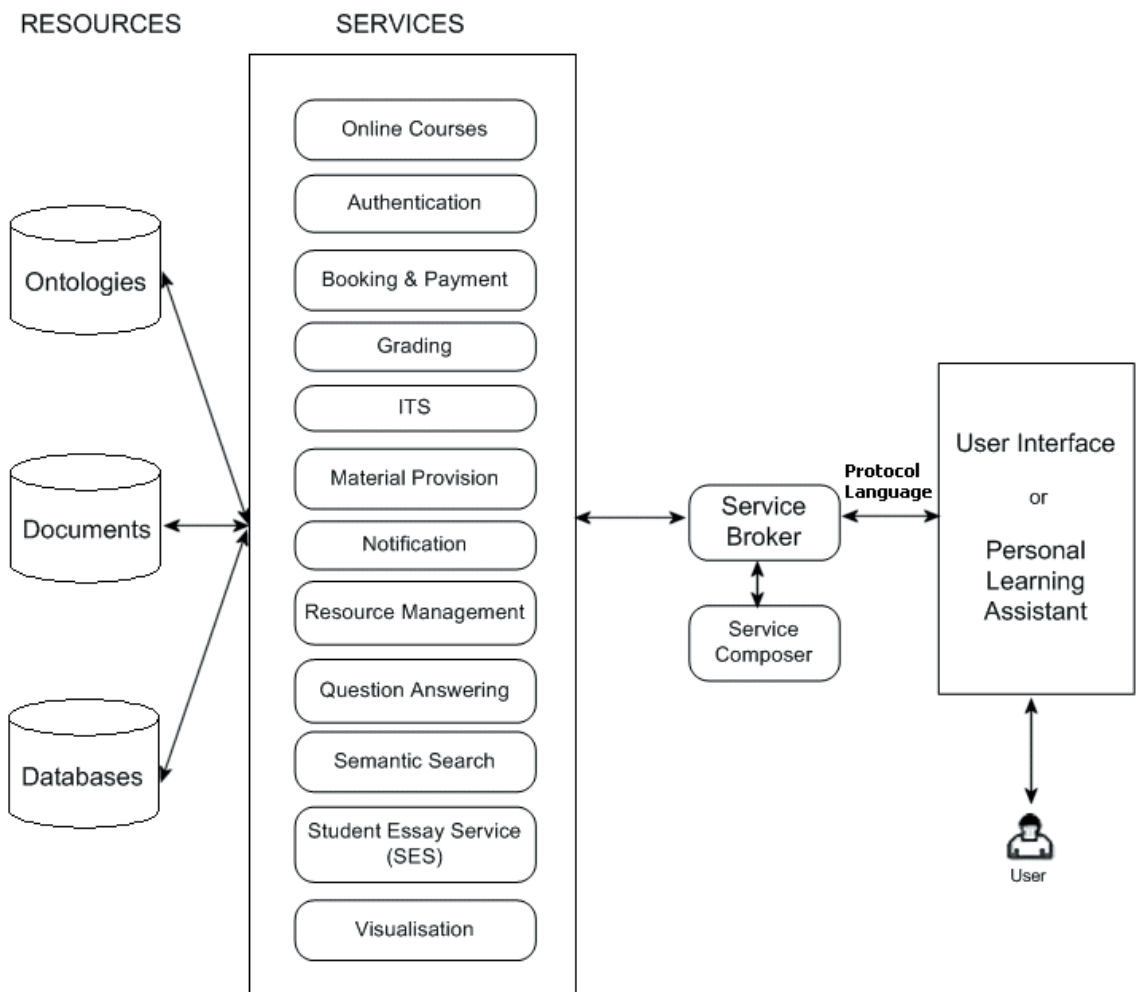


Figure 1 - Proposed Architecture for e-Learning Services

Scenario

To illustrate the architecture, we will now go through an e-learning scenario. A student first searches for an online course (optionally specifying any constraints): the broker handles the request and returns a set of choices satisfying the query. If no course is found, the user can register with a notification service. Otherwise, the user may find a suitable course among the offerings and then makes a final decision about registering for the course.

Processing the registration can be seen as a complex service involving registering with the system (resource management), creating a confirmation notification, creating a student account (authentication/authorisation), providing learning materials (provide materials) and processing payment (booking & payment), if applicable. Once all this is in place, the student can start the course. As part of the course, a student will be logging on and checking her learning agenda (e.g. next assignment due). This request is answered by combining several sources of information, such as course schedule, current date and student progress to date (e.g. completed units).

6. Conclusions and Discussion

We have devised our own categorisation of links of argumentation in student essays. This characterisation was performed by taking into consideration previous characterisations (of argumentation links) existing in the domain of research papers. We analysed some of the common patterns which can be found in a student essay. However, in our view, complex patterns are encountered which require higher order logics and, in particular, they require many-sorted logics.

We have tested several relational systems for information extraction, in particular RAPIER, BETH and FOIL. These systems extract restricted patterns handling unary types associated to variables/slots. None of these systems cope with the polymorphic type of each of the extracted arguments/slots. Their learning algorithms are restricted to learning either regular expressions or a restricted set of First Order Logic formulae with one type associated to the variables. However, in our domain of application, we require a system which can learn many-sorted formulae.

Finally, we discuss our vision of an agent-based student portal. This portal is illustrated by showing some of the semantic services which could be provided to a student in an e-learning scenario. The suggested architecture offers a set of semantic services such a smart question answering, intelligent information retrieval, a visualization tool for student essays, an exam marking tool, etc. Agents engage in retrieving the closest service which matches a student query (in some cases a complex service might be the combination of simple services). In our vision, agents can reason and perform intelligent tasks on behalf of a student. Then, the student could confirm that suggestions are acceptable (i.e. the solution offered satisfies all the constraints imposed by the student).

Acknowledgements

This work was funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

References

- Brill E. (1994). Some advances in rule-based part of speech tagging. In proceedings of the Twelfth National Conference on Artificial Intelligence, pages 722-727.
- Buckingham Shum, S., Uren, V., Li, G., Domingue, J., Motta, E. (2002) Visualizing Internetworked Argumentation, in *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Paul A. Kirschner, Simon J. Buckingham Shum and Chad S. Carr (Eds.), Springer-Verlag: London. <http://www.VisualizingArgumentation.info>
- Califf M.E. (1998). PhD thesis, Department of Computer Sciences, University of Texas, Austin, TX, August 1998.. Also appears as Artificial Intelligence Laboratory Technical Report AI 98-276 <http://www.cs.utexas.edu/users/ai-lab>
- Califf M.E. and Mooney R.J. (1997). In Proceedings of the ACL Workshop on Natural Language Learning, pages 9--15.
- Clocksini W. F. and Mellish C. S. (1981). *Programming in Prolog*, Springer-Verlag.
- Hyland, K. (1998) Persuasion and Context: The Pragmatics of Academic Metadiscourse, *Journal of Pragmatics* 30 (1998), 437-455.
- Lloyd J. W. (2000). A Logical Setting for the Unification of Attribute-Value and Relational Learning. In Proceedings of the workshop Attribute-Value and Relational Learning: Crossing the Boundaries in association with the Seventeenth International Conference on Machine Learning (ICML-2000). Stanford University.
- Mitchell T. (1997). *Machine Learning*. McGraw-Hill International.
- Mooney R.J., Melville P., Tang. R. L., Shavlik J., De Castro Dutra I., Page D. and Santos Costa V. (2004). Relational Data Mining with Inductive Logic Programming for Link Discovery. In *Data Mining. Next Generation Challenges and Future Directions*. Kargupta et al. (eds.) AAAI Press.
- Moreale E. and Vargas-Vera M. (2003). Genre Analysis and the Automated Extraction of Arguments from Student Essays. The Seventh International Computer Assisted Assessment Conference (CAA-2003). Loughborough University, 8-9 July.
- Moreale, E. and Vargas-Vera, M. (2004). Semantic Services in e-Learning: an Argumentation Case Study. *IFETS Journal of Educational Technology & Society - Special Issue on Ontologies and the Semantic Web for E-learning*, Volume 7 Issue 4 October.
- Muggleton S. (1995). Inverse entailment and Progol. *New Generation Computing Journal*, 13:245-286.

- Muggleton S and Feng C. (1992). Efficient induction of logic programs. In S Muggleton, editor, *Inductive Logic Programming*. Academic Press, New York, pages 281-297.
- Quinlan J. R and Cameron-Jones R. M. (1993). FOIL: A Midterm Report. *ECML 1993*: 3-20.
- Riloff, E. (1996). An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *AI Journal*, Vol. 85, August 1996.
- Swales, J.M. (1990) *Genre Analysis*. Cambridge University Press.
- Teufel, S., Carletta, J. and Moens M. (1999). An Annotation Scheme for Discourse-Level Argumentation in Research Articles, *Proceedings of EACL'99*, 110–117.
- Soderland S., Fisher D., Aseltine J., Lehnert W. (1995). Inducing a Conceptual dictionary. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*.
- Vargas-Vera, M. and Motta, E. (2004). AQUA - Ontology-based Question Answering System. *Third International Mexican Conference on Artificial Intelligence (MICAI-2004)*, Lecture Notes in Computer Science (LNCS 2972), Springer-Verlag, April 26-30.
- Vargas-Vera, M., Motta, E. and Domingue, J. (2003). AQUA: An Ontology-Driven Question Answering System. *AAAI Spring Symposium, New Directions in Question Answering*, Stanford University, March 24-26.
- Vargas-Vera M. and Celjuska D. (2004). Event Recognition on News Stories and Semi-Automatic Population of an Ontology. *IEEE/ACM International Joint Conference on Intelligent Agent and Web Intelligence (WI 2004)*, Beijing, China, IEEE Computer Society Press, 20-24 2004 September.
- Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A. and Ciravegna. F. (2002). MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. *The 13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, Lecture Notes in Computer Science 2473, ed Gomez-Perez, A., Springer Verlag, 2002, 379-391.
- Vargas-Vera M., Domingue J., Y. Kalfoglou Y., Motta E. and Buckingham-Shum S. (2001). Template-driven information extraction for populating ontologies. *Proc of the IJCAI'01 Workshop on Ontology Learning*, Seattle, WA, USA.
- Walther C. (1985). Unification in many-sorted theories. In *Proceedings of the 6th European Conference on Artificial Intelligence (ECAI-96)*, pp 593-602.
- Zelle J. M., Mooney J.R. and Konvisser J.B. (1994). Combining Top-down and Bottom-up Methods in Inductive Logic Programming, *Proc of The 11th International Conference on Machine Learning (ML-94)*, pp.343--351.