Lassoing Rhetoric with OWL and SWRL

Cliff O'Reilly and Shamima Paurobally

School of Informatics,
University of Westminster,
London,
W1W 6UW,
UK
cliff.oreilly@my.westminster.ac.uk,
s.paurobally@westminster.ac.uk

Abstract. We describe the LaRhetO application that takes a natural language text input and uses syntactic parsing tools to produce a knowledge base of linguistic entities using references to our own OWL ontological framework. Algorithms extend the knowledge base further to provide Meaning Representations of the text. We further expand the knowledge base to include references from external data sources such as DBPedia and thereby add world knowledge to the discovery. Finally the application locates rhetorical figures of speech and Rhetorical Structure Theory relations using our bespoke SWRL logic rules. Importantly our framework of ontology and rules as well as the application outputs form a part of the Semantic Web and Linked Data. We conclude that this research contributes to the Natural Language Processing (NLP) domain by automatically creating bespoke knowledge bases using Semantic Web tools that are published online and re-usable, but also by creating Meaning Representations useful for text analysis using syntactic parsing, Cue Phrase analysis and external world knowledge augmentation facilitated by Linked Data.

1 Introduction

In this paper we describe the analysis and development of a computer program for the automated discovery of figures of speech and rhetorical patterns within natural language. We call the computer program LaRhetO (Lassoing Rhetoric). The purpose of the project was two-fold: to develop a working prototype that could find various language patterns by using Semantic Web tools, and also to develop a knowledge base for each text input that would be augmented dynamically with information from the Semantic Web. We endeavoured to ensure that any mechanisms of inference beyond knowledge-gathering were re-usable as part of the Semantic Web movement.

1.1 Rhetoric

Since Classical times the structure of natural language has been analysed and documented with respect to forms of communication intended to persuade, teach,

enlighten, cajole, move or amuse. There are hundreds of documented rhetorical patterns of words; sometimes known as Figures of Speech. Many of these have been described since antiquity [11]. Rhetorical patterns vary from simple repetitions of words or phrases to discourse-level subtexts. Many are very difficult to discover automatically as they require complex knowledge such as context and world knowledge, Anaphora Resolution, Coreference Resolution and, in general, a detailed, almost innate, understanding of discourse (skills that human readers take for granted).

Rhetorical Structure Theory (RST) [12] is a Linguistic theory describing the organisation of natural language text. It characterises document structure in terms of Relations that hold between two non-overlapping spans of text called the Nucleus and the Satellite.

1.2 Meaning Representation

We think that one of the central aims of automated rhetorical analysis is to provide Meaning Representations - formal structures that capture the meaning of linguistic expressions. They have the potential to "bridge the gap from linguistic inputs to the non-linguistic knowledge of the world needed to perform tasks involving the meaning of linguistic inputs" [10]. The meaning of linguistic inputs causes problems for computer analysis of natural language in many areas such as question-answering and language translation.

Manual annotation of text is well documented. There has been some research and success into the automatic discovery of rhetoric [4], [14], but we believe the use of Semantic Web reasoning mechanisms to be new. This paper addresses the problem of a lack of an automated, computational rhetoric-discovery tool which uses Semantic Web technologies (OWL, SWRL and Linked Data). One approach to solving the problem is to furnish automated systems with more world knowledge. It is our view that, for the first time, a large amount of world knowledge is becoming readily computationally useful and available via the Semantic Web and Linked Data. Projects such as Wikipedia (and DBPedia) provide unprecedented amounts of data with good standards [9] and are in effect available free of charge, permanently and persistently.

2 LaRhetO

LaRhetO utilises tools and theories from Linguistics, together with the burgeoning resource of mechanisms and information that is the Semantic Web and Linked Data, in order to parse text and subsequently create and extend a knowledge base such that Meaning Representations are created. In this project we used the ontology editor Protégé from Stanford University. The most important aspect of Protégé for this project is that it has a programmable API in Java which we used from within the LaRhetO application. We also used the General Architecture for

Text Engineering (GATE) [5] with the Stanford Parser plug in. This is achieved via the GATE API in Java. The input used was the text block from the user and the output was the parsed natural language text marked up in XML.

2.1 Ontologies

We developed a suite of OWL ontologies for the various requirements of the project:

1. the GATE processing output, including the Stanford Parser - gate.owl
2. language entities and relationships - langtag.owl
3. higher level language/text entities and relationships - DocStruct.owl
4. external entities and relationships and associated intra-ontology relationships - VerbNounCombo.owl
5. rhetorical devices, figures of speech and
   RST relations - RhetoricalDevices.owl & RhetoricalStruct.owl

These ontologies are discussed in more detail below.

We used one OWL ontology to model the GATE output itself, e.g. Tokens, StartNodes, words, Dependencies etc, and a second OWL ontology for the specific language category tags generated e.g. WhAdverb, PastParticipleVerb etc. Other important objects and relationships needed to be modelled included Document, sentence, hasFirstWord, hasLastWord, hasNextWord, hasNextSentence etc. These elements were represented in the Document Structure OWL ontology which also contained a relationship to the hash code representation of the input text in order to identify uniquely the text that was input to the application. This was important if the output was to be re-used as it would enable the publication of the RDF triples of the knowledge base online. A representation of the Document Structure ontology is shown in Figure 1.

In order to classify different types of rhetorical forms an ontology of Rhetorical Devices was created. This included various instances of rhetorical devices, e.g. Adjunctio and Parelcon. This ontology also contained Linked Data resources for various types of language structure characteristics. The Rhetorical Devices ontology had the class of RhetoricalDevice. Related to the RhetoricalDevice class was the hasRhetoricalDevice object property. This OWL object property related the devices to the Doc instance in question and thereby allowed the population of the knowledge base with the knowledge that the device has been found. In order to specify the exact location of the pattern the GATE ontology object properties hasStartNode and hasEndNode were used. This enabled the user to pinpoint exactly where in the document any patterns had been found. This ontology also contained some more complex descriptive classes that were used to add more information to the knowledge base. For example when a word had a number of characters removed from its middle for effect this is classed as a MedialCut, for example “libary”. This was not classed as a misspelling but a deliberate play on words. Similarly for characters cut from the end of a word, such as “oft”, the word was classed as FinalCut. The EpitheticalName class was

used to hold specific names of people used in a rhetorical fashion. The use of instances of these classes contained in the OWL ontology enabled the processing mechanism to scan the input text for these particular patterns and tagged the resulting information in the knowledge base.

Our Lassoing Rhetoric OWL ontology imported these OWL files and also held the SWRL rules. It enabled a consolidated view of the knowledge base to be obtained from a single OWL file and also enabled relationships between the different ontological elements to be described more easily.

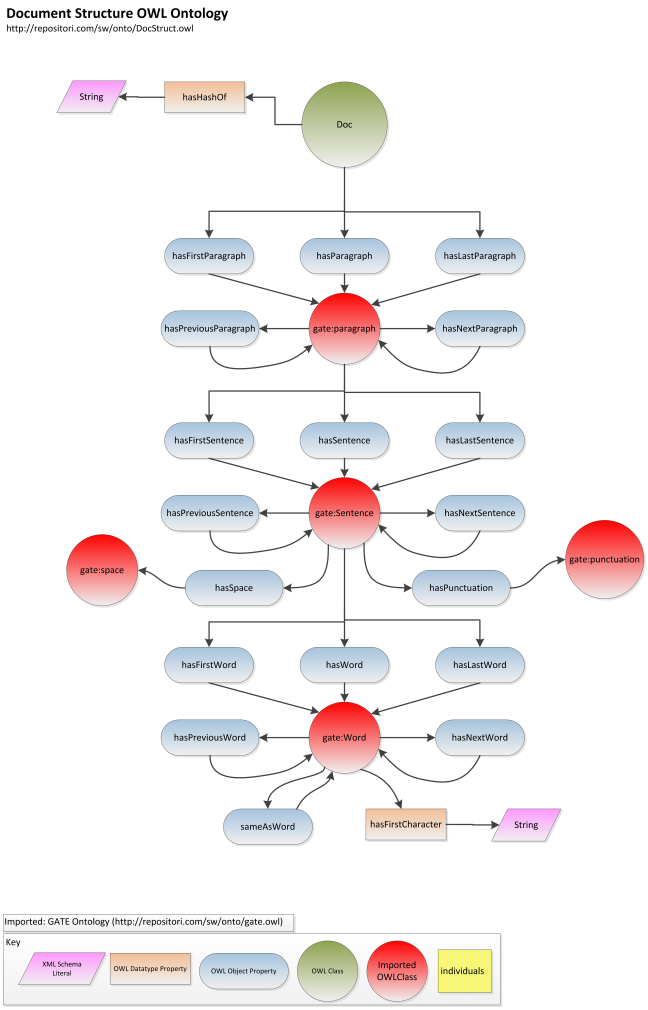

Fig. 1. Document Structure OWL ontology schematic representation

2.2 Populating the Knowledge Base

The Protégé-OWL API enabled the creation of a virtual knowledge base initially consisting of OWL classes and properties. The API was pointed to the OWL ontology LassoingRhetoric.owl for validation of the OWL entities. Step by step the GATE output was analysed and instances were created for the various classes and properties from the ontology. This occurs entirely in memory. At the stage where all the paragraphs, sentences, spaces and words with their associated properties such as hasWord or hasFirstSentence were added based on the GATE output (thereby the input text) we started on the creation of more knowledge based on external data. By reference to the ontologies created earlier of RhetoricalDevices and VerbNounCombo we were able to scan the text and add properties for the MinusComparison, PlusComparison EpitheticalName and SuperfluousWord OWL Classes. We were also able to scan the text for three adjacent words with the same initial letter. Since this might be a signal for the Alliteration rhetorical device we added the hasFirstCharacter datatype property for relevant instances of the word OWL Class. The next step was to refer to external knowledge bases not previously created by us. The British National Corpus is an online resource of word frequencies derived from thousands of source texts. We were able to remotely query this resource with a particular combination of noun and verb words selected from our text in order to detect previous usage of the combination. If the British national Corpus had no record for the query then we were able to add the UnusualNounInCombo and UnusualVerbInCombo properties to our knowledge base for the two word instances in question. This implied that the grouping was unusual and could indicate the presence of the rhetorical form of Catachresis (to which "Lassoing Rhetoric" could be said to belong).

A further Linked Data analysis was undertaken based on the DBPedia online resource. The rhetorical form of Historic Present can be signalled by a present tense reference to a person no longer living. By querying DBPedia dynamically with the name of the individual from the text we were able to determine whether firstly they were a person at all i.e. did they have a birthdate property and secondly are they alive at the application runtime, i.e. did they have a deathdate property. The resulting records were then used to populate the knowledge base with a relation to the Person OWL Class from the DBPedia ontology, and also whether alive or dead - AlivePerson OWL Class disjoint with DeadPerson OWL Class from our Lassoing Rhetoric OWL ontology. Another example of a Linked Data query we were able to produce was the information extracted from WordNet. WordNet is another Linked Data resource, but related to the English language entities and their synonyms, antonyms and various other characteristics that words have. Whenever we encountered a verb word in the input text we were able to query the online presence to have returned the synsetID. The synsetID is a unique reference to the group of synonyms to which the particular word belongs, for example the verb "to have" belongs to the synonym set of "have, have got, hold". The same word can also belong to different synonym sets, however for our purposes it was enough to be able to return a single syn-

onym set and then query a local copy of Extended WordNet[1] to furnish the application with the synonyms of the word in question. After all these various analyses the virtual knowledge base was complete for our purposes. It was then possible to execute the SWRL rules dynamically. Our program was enabled so that the report file that was output would contain descriptions of each rhetorical pattern located. The entire knowledge base was then output to an RDF file which contained all the original ontology data (from LassointRhetoric.owl) and also all the new individuals populated from the processing. Crucially this RDF file was to be uploaded to an online location. The architecture of LaRhetO is shown in Figure 2. Once we had the knowledge base populated and the SWRL rules were run we were able to cycle through each inferred axiom and insert it into the output text with some HTML and Javascript that enabled us to display the original text together with coloured spans showing the rhetorical patterns discovered.

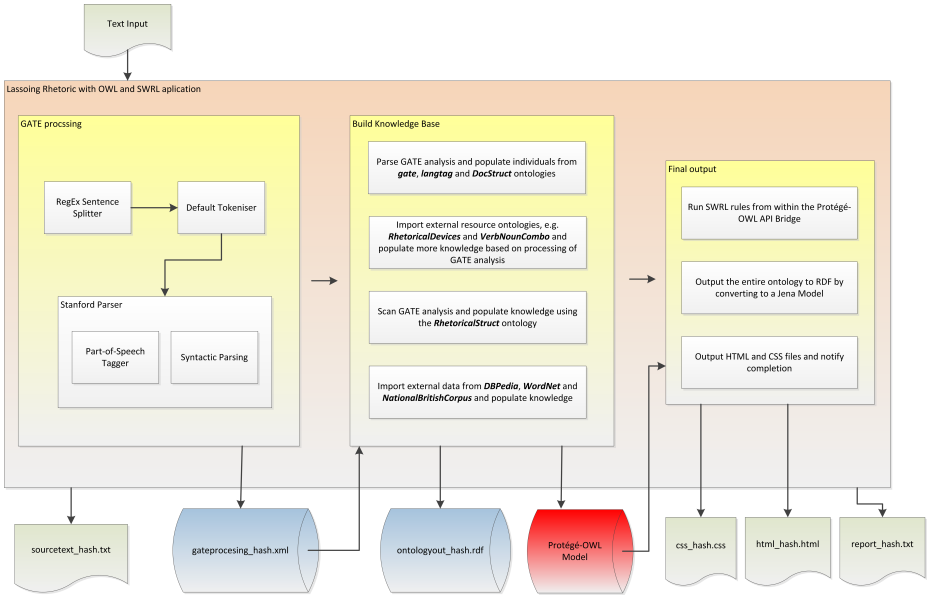

Fig. 2. LaRhetO architecture

2.3 Rhetorical forms and logic rules

Anaphora is an example of a rhetorical form. It occurs when there is "repetition of the same word at the beginning of successive clauses or versus" [11]. An example is found in Winston Churchill's speech to Parliament in 1940 when

[1] http://xwn.hlt.utdallas.edu/

he uses the phrase "We shall" at the beginning of various clauses, for example "We shall fight them on the beaches". In order to discover this pattern in text using a knowledge base and logic rules we needed a number of entities namely a Document, a paragraph, a sentence and at least two clauses. We also need the entities to be related such that a clause that has an adjacent clause and has the same first two words (by orthography) can be related to a sentence and also a parent paragraph and then to a parent Document. This is a very basic logical test and can be summarised by the following syllogism:

IF Document HAS Paragraph
AND IF Paragraph HAS Sentence
AND IF Sentence HAS Clause A
AND IF Sentence HAS Clause B
AND IF Clause A HAS Word X
AND IF Clause A HAS Word Y
AND IF Clause B HAS Word F
AND IF Clause B HAS Word G
AND IF Word X IS THE SAME AS Word F
AND IF Word Y IS THE SAME AS Word G
THEN Document HAS Anaphora

We assumed for this rule that the meaning of the word is irrelevant: at its simplest the repetition of the same orthographic form signals Anaphora. An Anaphoric pattern could, of course, extend to more than two matching clauses and vary over more than two adjacent clauses also, however for this project we were only concerned with developing a single rule for each rhetorical pattern selected. LaRhetO was not intended to be a complete rhetorical device locater. This syllogism can be converted to Horn Clauses quite easily and therefore we were able to generate a suite of SWRL rules as Horn Clauses based on the logic of discovering new knowledge from the knowledge base built up for the text under analysis. The resultant SWRL rule for Anaphora was:

$$
\begin{aligned}
&DocStruct{:}Doc(?h) \wedge DocStruct{:}hasParagraph(?h, ?i) \wedge \\
&DocStruc{:}hasSentence(?i, ?z) \wedge gate{:}word(?x) \wedge \\
&DocStruct{:}hasNextWord(?x, ?y) \wedge gate{:}Sentence(?z) \wedge \\
&DocStruct{:}hasFirstWord(?z, ?x) \wedge gate{:}word(?a) \wedge \\
&DocStruct{:}hasNextWord(?a, ?b) \wedge gate{:}Sentence(?c) \wedge \\
&DocStruct{:}hasFirstWord(?c, ?a) \wedge DocStruct{:}hasNextSentence(?z, ?c) \wedge \\
&gate{:}hasString(?x, ?d) \wedge gate{:}hasString(?y, ?e) \wedge gate{:}hasString(?a, ?f) \wedge \\
&gate{:}hasString(?b, ?g) \wedge swrlb{:}equal(?d, ?f) \wedge swrlb{:}equal(?e, ?g) \wedge \\
&gate{:}hasStartNode(?x, ?j) \wedge gate{:}hasEndNode(?b, ?k) \rightarrow \\
&RhetDev{:}hasRhetoricalDevice(?h, RhetDev{:}Anaphora) \wedge \\
&gate{:}hasStartNode(RhetDev{:}Anaphora, ?j) \wedge \\
&gate{:}hasEndNode(RhetDev{:}Anaphora, ?k)
\end{aligned}
$$

The pre-existing instance of RhetDev:Anaphora was used in the consequent to create an object property relationship to the Doc instance. The Anaphora object was then related to start and end nodes for future use within the tool. All our

rules were validated using Protégé against the Lassoing Rhetoric OWL ontology. Since SWRL is not a productive rule system it was not possible to have instances generated from a rule, therefore all variables in the rule must refer to existing entities or literals. The effect of this was that in order to have rhetorical forms "discovered" an entity must already exist at the time of inference, i.e. be a part of the knowledge base when the rule is executed. The various instances for the RhetoricalDevice class were contained in the Rhetorical Devices ontology. These were used in the consequent of each rule and came into play wherever all the atoms in the antecedent of the rule were true.

We also assume that text input is well formed and correct, e.g. in order for our rule to find successfully the form of Catachresis we have to assume that the verb/noun combinations are intentional, e.g. shave grass / lasso rhetoric etc. Any mistaken text input could easily be misconstrued as rhetorical even though it is not. This is a complex issue, however. An example is the rhetorical device of Enallage. This works by deliberate mistakes being taken at face value for rhetorical effect, e.g. the use of the word "wot" instead of "what".

2.4 Outputs

The final output of this project was an application that allows input text to be uploaded and outputted a report containing information about the processing of the text and details of where various files were stored, e.g. output RDF. The input text was also displayed marked-up with any rhetorical forms that were found by the tool. Notes and background information regarding the particular rhetorical patterns were provided to give valuable information to users. We enhanced the information output and benefits to the Semantic Web by publishing the knowledge base online for each text input. This included an automatically-created RDF output of the Protégé-OWL API Bridge - this is the entire knowledge base; a text file log of the processing steps and comments of the program as it ran through the analysis; the original source text in a txt file; and the output of the GATE analysis in XML format. All these output files were named using the unique hash code generated automatically by the program at execution. By storing all these documents and notifying the user of the hash code and the online location of these files they can be re-used within the Semantic Web framework.

3 Related Works

Text technologies, Discourse Parsing, RST, figures of speech analysis, ontologies and the Semantic Web have many years of research history spanning different domains, e.g. in [1], [7], [11], [13]. All of these fields have rarely been combined. Bärenfänger et al [2] developed an ontological approach to discourse parsing with RST. They produced two ontologies, a novel RST taxonomy and the GermaNet German lexicon both using OWL DL ontologies. Using Prolog rules, German text is analysed through a multilayer, iterative parser in order to annotate the

text for RST relations. Also attempting to parse RST elements Corston-Oliver [4] describes a discourse analysis component within the Microsoft English Grammar (MEG) system called Rhetorical Structure Theory Analyzer (RASTA). He discusses various methods of recognising discourse relations, e.g. using references to world knowledge, reasoning with representations of propositional content and reliance on cue words, phrases or lexical repetition. The last method is preferred in this paper since the other two have questions that are "by no means resolved".

Graham Wilcock [15] combines the Semantic Web tools OWL and SWRL with NLP. He shows that it is possible to categorise phrases and sentences using grammar rules and an ontology framework. Cimiano and Reyle [3] specifically research meaning representations in combination with an ontological/semantic approach to NLP.

Two other papers that report ontology-based annotation research are [6] and [8]. Gawyjołek's research successfully discovers rhetorical forms very similar to this paper, however he does not utilise Semantic Web technology and re-usable knowledge bases.

Our work shares many similarities with the papers detailed above. However, we propose new research due to the nature of the rhetorical patterns discovered, e.g. Classically-termed figures of speech, and the combination of OWL, SWRL and dynamic, external world knowledge augmentation.

4 Conclusion and further work

In this paper we report the LaRhetO application that discovers rhetorical forms from natural language. It does so within the Semantic Web domain by utilising OWL, SWRL and Linked Data, but also by publishing both the outputs of each text block entered as a knowledge base and the ontological framework and SWRL rules online. The rules we generated are not complete; it cannot be said that all instances of the rhetorical forms would be found by the rules designed herein. This is, however, to be expected. Natural language and rhetorical devices are complex forms of communication and we cannot expect to pinpoint them all with simple logic rules.

The ultimate purpose of this project is to disambiguate natural language such that a computer can formulate meaning representations from it. It is our belief that we have gone some way towards that aim. In comparing our project with the related works it seems clear that the basis of language parsing and analysis coupled with rhetorical pattern discovery is the same. Where this project differs is in attempting to address the issue of world knowledge inference. The related papers mention world knowledge, but often imply that the problem is too difficult to address. The Semantic Web provides a mechanism to chip away at this problem and as the domain of Linked Data grows it will become possible to augment dynamically and persistently NLP analyses with world knowledge considered previously to be too difficult.

Since this project spans a number of domains the contribution it makes is (at least) two-fold. Firstly, in the field of linguistics the development of knowledge

bases upon which to base language analyses is not new. Even the relatively new Semantic Web tools (e.g. OWL ontologies) have been used previously in his field. It is a powerful mechanism that enables effective investigation and persistency of results (via OWL ontology publication). We have taken the same view and utilised the GATE suite of algorithms and OWL ontologies. We feel that the outputs of the textual analysis are valuable because they add knowledge that was not previously available and in a format that is reusable and published online thereby enabling greater accessibility and expandability. Linguistic studies have also used logical reasoning previously. We take a similar approach, but have tried to stay within the Semantic Web domain by using SWRL. Our contribution then has been to analyse text and publish the resultant knowledge base online and as an input to the Semantic Web. It is possible for a resultant RDF file to be referenced in future studies in many ways including dynamic querying and expansion. Secondly, the process of adding world knowledge to this kind of analysis is not completely new. The mechanism that we have shown to work is based on sound computing premise - the Semantic Web. This resource is blossoming and with an appropriate framework in place can add knowledge to investigations of meaning that previously would not be possible to automate.

Perhaps more importantly for our purpose we have shown that the development of Meaning Representations within our knowledge base by virtue of the Description Logic and Horn Logic entities (within our OWL ontologies and SWRL rules respectively) is possible with Semantic Web technologies and that the dynamic addition of world knowledge can increase the potential for Meaning Representation-based inference and analysis. In our time computers cannot "understand" natural language. This is a significant problem that is being addressed in many ways. We argue in our project that the Semantic Web and Linked Data provide a mechanism to solve this problem somewhat. We believe that with the recent advances in computing technologies, specifically the Semantic Web, that the paradigm of knowledge base inference can be extended usefully to include world knowledge for the first time. We show that it is possible even if in a limited manner. The data sources available online are growing rapidly, but at the moment are not intended for augmenting Meaning Representations and linguistic analyses. Over time we expect that more data will become available that is more easily integrated into this kind of processing of language and that perhaps greater ontological frameworks will be developed that will allow highly complex knowledge bases, rules and inference to work in this important and expanding domain.

Further work in these domains could progress along many lines, however we mention the following:

1. Whilst we have been able to prove our thesis we find that SWRL is limited in its potential due to the facts that it is not computationally complete, doesn't support negated atoms and also that it is not productive. We think that SWRL has a very useful position within the Semantic Web, however for our purposes we would prefer to use a different rule system - one that we haven't seen yet. With the advent of the RIF framework we will expect

to see more rule systems being developed for specific purposes. Our view is that a linguistic-based rule system that has the capabilities of SWRL, but also enables entity-creation and computational completeness and, crucially, within the Semantic Web domain is required for future work in this area
2. Statistical linguistic analyses are mature and would prove beneficial to future work in this area by providing probability breakdowns for hard-to-decide algorithm outputs. For example trying to determine if a word is simply misspelt or is deliberately used in this way is very difficult. Humans find it easy to Code-Shift between dialects or languages, but a computing system cannot easily do this. A machine-learning approach would add significant benefits to this domain.
3. Various other data sources on the Semantic Web were investigated for this project, however not many of them provide the kinds of information easily obtainable that would benefit this simple knowledge base we were creating. Sources such as SUMO, YAGO, Cyc and Freebase provide excellent resources, but without having undertaken further research into this problem it was not possible to use them as resources at that stage. We expect that these and many other data sources could be investigated, made interoperable and utilised for similar lines of research.

References

1. Abulaish, M., Using Part-Of-Speech Patterns and Domain Ontology to Mine Impreceise Concepts frm Text Documents. Proceedings of the 6th International Conference on Information Integration and Web Based Applications and Services (iiWAS'04). Jakarta, Indonesia. Sept. (2004)
2. Bärenfänger, M., Hilbert, M., Lobin, H., Lüngen, H.: OWL ontologies as a Resource for Discourse Parsing. LDV-Forum. GLDV-Journal for Computational Linguistics and Language Technology. 23(1):17-26. (2008)
3. Cimiano, P., Reyle, U.: Towards Foundational Semantics - Ontological Semantics Revisited Proceedings of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006). Baltimore, USA. Nov. 2006.
4. Corston-Oliver, S.H.: Beyond String Matching and Cue Phrases: Improving Efficiency and Coverage in Discourse Analysis. Technical Report - American Association for Artificial Intelligence SS No. 06:9-15. (1998)
5. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.
6. Erdmann, M., Maedche, A., Schnurr, H.P., Staab, S.: From Manual to Semiautomatic Semantic Annotation: About Ontology-based Text Annotation Tools Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, 2000. Saarbrücken, Germany. July, 2000.
7. Estival, D., Nowak, C., Zschorn, A.: Towards Ontology-Based Natural Language Processing Proceedings of the NLPXML '04: Proceeedings of the Workshop on NLP and XML (NLPXML-2004): RDF/RDFS and OWL in Language Technology. Barcelona, Spain. July, 2004.

8. Gawryjołek, J., DiMarco, C., Harris, R.: Automated Annotation and Visualization of Rhetorical Figures Computational Models of Natural Argument (CMNA). Pasadena. July, 2009.
9. Giles, G.: Internet encyclopaedias go head to head. Nature, 438:900-901. (2005)
10. Jurafsky, D. and Martin, J.: Speech and Language Processing. 2nd ed. New Jersey: Pearson Education Inc. (2009)
11. Lanham, R.A.: A Handlist of Rhetorical Terms. Berkeley: University of California Press. (1991)
12. Mann, W. C. and Thompson, S. A.: Rhetorical Structure Theory: Toward a functional theory of text organisation. Text, 8(3):243-281. (1988)
13. Pardal, J.: Dynamic Use of Ontologies in Dialogue Systems Proceedings of the NAACL-HLT 2007 Doctoral Consortium. pages 25-28. Rochester, NY, USA., April 2007.
14. Reitter, D.: Simple Signals for Complex Rhetorics: On Rhetorical Analysis with Rich-Feature Support Vector Models. Journal for Computational Linguistics Language Technology. 18(1/2):38-52. (2003)
15. Wilcock, G.: Natural Language Parsing with GOLD and SWRL. In Online Proceedings of the RuleML Conference, Rules and Rule Markup Languages for the Semantic Web, Second International Conference, RuleML, Athens, Georgia. Nov. 2006