Enacting textual entailment and ontologies for automated essay grading in chemical domain

Adrian Groza and Roxana Szabo
Intelligent System Group,
Department of Computer Science,
Technical University of Cluj-Napoca, Romania
Adrian.Groza@cs.utcluj.ro,Roxana.Szabo@cs-gw.utcluj.ro

***Abstract*—We propose a system for automated essay grading using ontologies and textual entailment. The process of textual entailment is guided by hypotheses, which are extracted from a domain ontology. Textual entailment checks if the truth of the hypothesis follows from a given text. We enact textual entailment to compare students answer to a model answer obtained from ontology. We validated the solution against various essays written by students in the chemistry domain.**

***Index Terms*—automated essay grading, natural language processing, textual entailment, ontologies**

I. Introduction

Assessment is an essential part of the learning process, especially in formative learning settings. In the current context of massive open online courses (MOOC), assessment is challenging as it aims to ensure consistency, reliability and do not favor one person against another. In formative assessment the problem of workload and timely results is even greater, as the task is carried out more frequently while the interpretation of one human marker differs from another.

While essay questions are advantageous to student learning and assessment there are obvious disadvantages for the instructor. Grading of essay and discussion questions is time consuming even with the help of teaching assistants. Automated essay grading [20] aims at automatically assigning a grade to a students essay by means of various features. Since the argument structure is crucial for evaluating essay quality, persuasive essays are extensively studied [21]. By automatically identifying arguments, the evaluator is be able to inspect the essay's plausibility. We argue that information technology is able to assist and support teachers in these challenges.

Our research hyphotesis relies on the correlation between textual entailment [1] and answer correctness. In a typical answer assessment scenario, we expect a correct answer to entail the reference answer. However a student may wish to skip the details already mentioned in the question. Hence, the problem is whether the answer, along with the question, entail the reference answer.

Let the question be a, student answer be s and the reference answer be r. Correctness means $a \wedge s \Rightarrow r$ and contradiction means $s \wedge a \Rightarrow \neg r$. We propose the usage of recognizing textual entailment (RTE) along with shallow text features to train on the system dataset and testing it on the test dataset provided. The evaluation metrics used will be according to the Coh-Metrix system [10]. The final grade will be obtained from the first grade computed from the comparison between the students answer and the hypotheses generated from the model ontology, and the second grade that will be obtained from evaluation of the metrics.

The rest of the paper is organized as follows: Section II presents the system architecture and the NLP tools enacted. Section III details a running scenario on students essays in the chemical domain. Section IV discusses related work, while section V concludes the paper.

II. System architecture

The proposed system consists of a set of integrated natural language tools (see Fig. 1). The system is structured on layers. The first layer contains the Text2Onto tool [3], used to obtain a consistent ontology from a corpus of high ranked or relevant essays in a given domain. The second layer exploits the OWL-Natural service [9], to generate natural text from a selected ontology. We organise the text generated by OWLNatural as a set of hypotheses.

In the second layer, textual entailment is used to analyse the domain hypotheses on the available essays. The EOP system [14] is trained using a set of pairs $\langle Text, Hyphotesis\rangle$. For experiments, we created a data set in the chemical domain containing 100 pairs of text/hypothesis divided into 50"%" of entailment pairs and 50"%" of non-entailment pairs. The text is represented by the student's essay to be reviewed. The hypotheses are generated from a domain ontology and filtered by the teacher. Based on the model generated after training, the EOP system computes the confidence of hypotheses entailment within the text. This confidence constitutes the basis for grading the essay.

Automatic grading includes also various readability metrics. For this step, we use GATE tool for natural language processing [4], to get the number of tokens or number of sentences from text. We integrate Coh-Metrix service [11] to compute various cohesion and coherence metrics for written texts.

The system components and the main workflow appear in Fig. 1. The following four components are detailed: (1) developing the domain ontology, (2) generating hypothesis from ontology, (3) textual entailment methods, and (3) natural language processing of essays.

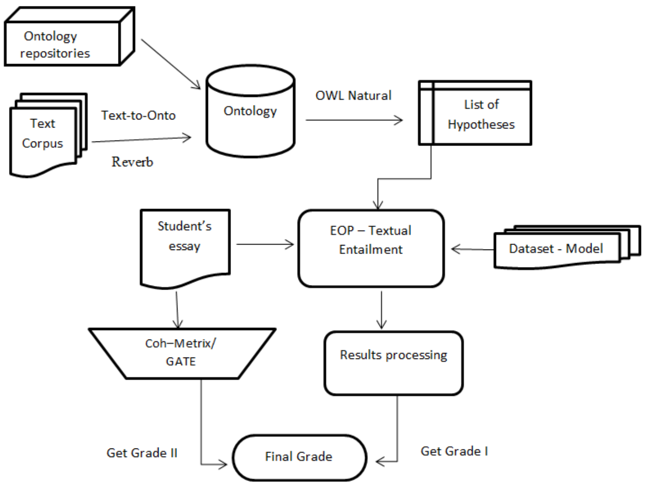

Fig. 1. System architecture.

A. Developing the domain ontology

We assume that the professor provides a corpus of relevant documents in the domain of interest. Our approach is to automatically generate a domain ontology from this corpus. For this task, we rely on Text2Onto framework for ontology learning from textual resources.

Three main features distinguish Text2Onto: Firstly, learned knowledge is represented at a meta-level Probabilistic Ontology Model (POM). Secondly, user interaction is a core aspect of Text2Onto and the fact that the system calculates a confidence for each learned object allows to design visualizations of the POM. Thirdly, by incorporating strategies for data-driven change discovery, we avoid processing the whole corpus from scratch each time it changes. Instead, POM selectively updates itself, according to the corpus changes only. Besides increasing efficiency, this solution allows to trace the evolution of the ontology with respect to the changes in the underlying corpus.

Text2Onto combines machine learning approaches with basic linguistic processing such as tokenization or lemmatizing and shallow parsing [**?**]. Since it is based on the GATE framework it is very flexible with respect to the set of linguistic algorithms used. Another benefit of using GATE is the seamless integration of JAPE rules which provides finite state transduction over annotations based on regular expressions. Linguistic preprocessing in Text2Onto starts by tokenization and sentence splitting.

The main workflow of ontology generation consists of: preprocessing, execution of algorithms, combining results. During preprocessing, Text2Onto calls GATE applications to tokenize the document, split sentences, tag Part of Speech and match JAPE rules. GATE creates indexes for the document and the result of this is obtained as an AnnotationSet. In the next step, Text2Onto executes the applied algorithms in a pre-specified order: i) concept, ii) instance, iii) similarity, iv) subclass-of, v) instance-of, vi) relation and vii) subtopic-of. The basic heuristic employed in Text2Onto to extract concepts and instances is that nouns represent concepts and proper nouns are instances. If more than one algorithm is applied for each category, then the final relevance value is computed based on the selected combiner strategies [12].

Given a student essay, we need to analyse its content against the domain ontology available from the previous step. We rely on ReVerb [6] to extracts triplets from the student essay.

ReVerb is designed for Web-scale information extraction, where the target relations cannot be specified in advance and speed is important. ReVerb first identifies relation phrases that satisfy the syntactic and lexical constraints, and then finds a pair of NP arguments for each identified relation phrase. The resulting extractions are then assigned a confidence score using a logistic regression classifier.

This algorithm differs in three important ways from previous Open IE systems like TextRunner [23]. Firstly, the relation phrase is identified holistically rather than word-by-word. Secondly, potential phrases are filtered based on statistics over a large corpus (the implementation of our lexical constraint). Finally, ReVerb is "relation first" rather than "arguments first", which avoids a common error like confusing a noun in the relation phrase for an argument, (e.g. the noun "deal" in "made a deal with") [6].

Given an input sentence s, ReVerb uses extraction algorithm 1. In the second part of the algorithm, for each relation phrase r identified in Step 1, find the nearest noun phrase x to the left of r in sentence s such that x is not a relative pronoun or the existential there. Then, the algorithm finds the nearest noun phrase y to the right of r in sentence s. If such an (x, y) pair could be found, the tuple $\langle x, r, y\rangle$ is returned.

```
Input : $s$ - sentence;
foreach verb $v \in s$ do
    find the longest sequence of words $rv$ such that:
    (1) $rv$ starts at $v$,
    (2) $rv$ satisfies the syntactic constraint, and
    (3) $rv$ satisfies the lexical constraint
end
if $\exists$ pair of matches adjacent or overlap in $s$ then
    merge them into a single match;
end
foreach relation phrase $r$ do
    find the nearest noun phrase $x$ to the left of $r \in s$ such
    that $x \notin RelativePronoun \cup \{There\}$,
    Find the nearest noun phrase $y$ to the right of $r \in s$.
end
return $\langle x, r, y\rangle$.
```

Algorithm 1: Relation extraction algorithm.

Open Information Extraction (IE) is the task of extracting assertions from massive corpora without requiring a pre-specified vocabulary. ReVerb takes raw text as input, and outputs triplets $\langle argument1,\ relationPhrase, argument2\rangle$, as illustrated in example 1.

Example 1 (Triplets extraction with ReVerb)**.** *Given the sentence: "Vitamin D is toxic in large amounts.", the extracted*

Every OrganicSulfurCompound $\equiv$ OrganicCompound $\sqcap$ $\exists$ hasPart . OrganicSulfurGroup.

H_1	Every OrganicSulfurCompound is an OrganicCompound that hasPart an OrganicSulfurGroup.
H_2	Every OrganicCompound that hasPart an OrganicSulfurGroup is an OrganicSulfurCompound.

Fig. 2. Generating natural language hypothesis from chemical ontology.

triple is: $\langle vitamin\ d,\ be\ toxic\ in,\ large\ amount\rangle$*. For "Bananas are an excellent source of potassium", ReVerb extracts the triple* $\langle bananas,\ be\ source\ of,\ potassium\rangle$.

B. Generating hyphothesis

We use OWLNatural to generate natural language hypotheses from a domain ontology. OWLNatural is natural language generation engine that produces descriptions of individuals and classes in English and Greek from ontologies that have been annotated with linguistic and user modeling information expressed in RDF. The OWL verbalizer takes its input in OWL syntax and produces an output in a fragment of Attempto Controlled English (ACE) [8].

C. Enacting textual entailment

Textual entailment (TE) is a directional relation between text fragments. The relation holds whenever the truth of one text fragment follows from another text. Given two text fragments, one named Text (T) - the entailing and the other named Hypothesis (H) - the entailed.

Recognizing Textual Entailment (RTE) has been proposed [5] as a generic task that captures major semantic inference needs across many natural language processing applications. The Recognizing Textual Entailment task consists in recognizing whether the Hypothesis can be inferred from the Text. We use a graduated definition of entailment: T entails H ($T \Rightarrow H$) if, typically, a human reading T would infer that H is most likely true. Positive entailment is illustrated in example 2.

Example 2 (Positive entailment)**.**

T: *In chemical reactions with metals, nonmetals gain electrons to form negative ions.*

H: *The nonmetals become negative ions.*

The correctness means that the text entails the hypothesis, so we obtain the positive entailment. The contradiction means that the text does not entail the hypothesis, and there are an negative entailment. An example of a negative TE (text contradicts hypothesis) is illustrated by example 3.

Example 3 (Negative entailment)**.**

T: *Nonmetallic elements also react with other nonmetals, in this case forming molecular compounds.*

H: *Metals react with nonmetals in order to form ions.*

An example of a non-TE (text does not entail nor contradict) is illustrated by example 4.

Example 4 (Non entailment)**.**

T: *A chemical reaction is one in which the organization of the atoms is altered.*

H: *The burning of methane is a chemical reaction is in the presence of oxygen.*

The Excitement Open Platform (EOP) is a generic architecture for textual inference in multiple languages. The platform includes state-of-art algorithms, a large number of knowledge resources, and facilities for experimenting. The input consists of the text T and hypothesis H. The output is an entailment judgment, either Entailment if T entails H, or NonEntailment if the relation does not hold. A confidence score for the decision is also returned in both cases.

The overall structure consists of two main parts: Linguistic Analysis Pipeline and Entailment Core. The Linguistic Analysis Pipeline (LAP) is a series of linguistic annotation components range from tokenization to part of speech tagging, chinking, Named Entity Recognition and parsing. Entailment Core consists of Entailment Decision Algorithms (EDAs) and more subordinate components. An EDA takes an entailment decision while components provide static and dynamic information for the EDA. The Entailment Decision Algorithm (EDA) computes an entailment decision for a given Text/Hypothesis pair, and can use components that provide standardized algorithms or knowledge resources. Currently, the EOP ships with three EDAs each following a different approach: transformation-based, edit-distance based, and classification based. Scoring Components accept a Text/Hypothesis pair as an input, and return a vector of scores. Distance Components that can produce normalized and unnormalized distance/similarity values in addition to the score vector. Annotation Components can be used to add different annotations to the Text/Hypothesis pairs. Syntactic Knowledge Components capture entailment relationships between syntactic and lexical-syntactic expressions.

Knowledge is crucial to recognize cases where T and H use different textual expressions (words, phrases) while preserving entailment (e.g., home → house, Hawaii → America, born in → citizen of). The EOP contains a wide range of knowledge resources, including lexical and syntactic resources. Part of them are mannually grabbed from dictionaries, while others are automatically learned. The EOP platform includes three different approaches to RTE: i) an EDA based on transformations between T and H; ii) an EDA based on edit distance algorithms; and iii) a classification based EDA using features extracted from T and H.

Transformation-based EDA applies a sequence of transformations on T with the goal of making it identical to H. Consider the following example where the text is The boy was located by the police and the hypothesis is "The child was found by the police". Two transformations: $boy \rightarrow child$ and $located \rightarrow found$ do the job.

Edit distance EDA involves using algorithms casting textual entailment as the problem of mapping the whole content of T

TABLE I
RESULT FOR THE ENTAILMENT DECISION ALGORITHM.

EDA	*Accuracy*
Transformation-based English RTE-3	67.13”%”
Transformation-based English RTE-6	49.55”%”
Edit-Distance English RTE-3	64.38”%”
Edit-Distance German RTE-3	59.88”%”
Edit-Distance Italian RTE-3	63.50”%”
Classification-based English RTE-3	65.25”%”
Classification-based German RTE-3	63.75”%”
Median of RTE-3 (English) submissions	61.75”%”
Median of RTE-6 (English) submissions	33.72”%”

into the content of H. Mappings are performed as sequences of editing operations (i.e., insertion, deletion and substitution) on text portions needed to transform T into H, where each edit operation has an associated cost. The underlying intuition is that the probability of an entailment relation between T and H is related to the distance between them.

Classification based EDA uses a maximum entropy classifier to combine the outcomes of several scoring functions and to learn a classification model for recognizing entailment. The scoring functions extract a number of features at various linguistic levels (bag-of-words, syntactic dependencies, semantic dependencies, named entities) [14]

MaxEntClassificationEDA is an Entailment Decision Algorithm (EDA) based on a prototype system called Textual Inference Engine (TIE). Results for the three EDAs included in the EOP platform are reported in Table I. Each line represents an EDA, the language and the dataset on which the EDA was evaluated.

D. Natural language processing with GATE.

For natural language procesing we use GATE (General Architecture For Text Engineering). In GATE the logic is arranged in modules that are called pipelines. GATE contains an information extraction pipeline called ANNIE composed of several components: Tokenizer, Gazetteer List,Sentence Splitter, POS Tagger, Semantic Tagger that annotates entities such as Person, Organization, Location, and an Orthographic Co-reference that adds identity relations between the entities annotated by the Semantic Tagger. The tokeniser splits the text into very simple tokens such as numbers, punctuation and words of different types. The role of the gazetteer is to identify entity names in the text based on lists.

III. RUNNING SCENARIO

Consider the essay in example 5.

Example 5 (Sample essay in the chemical domain)**.**

“All the matter in the universe is composed of the atoms of more than 100 different chemical elements, which are found both in pure form and combined in chemical compounds.First,a sample of any given pure element is composed only of the atoms characteristic of that element, and the atoms of each element are unique. For example, the atoms that constitute carbon are different from those that make up iron, which are in turn different from those of gold. Every element is designated by a unique symbol consisting of one or more letters arising from either the current element name or its original Latin name. For example, the symbols for carbon ,hydrogen, and oxygen are simply C, H, and O, respectively. The symbol for iron is Fe, from its original Latin name ferrum. On the other hand, the chemical compound is any substance composed of identical molecules consisting of atoms of two or more chemical elements. The fundamental principle of the science of chemistry is that the atoms of different elements can combine with one another to form chemical compounds.Methane, for example, which is formed from the elements carbon and hydrogen in the ratio four hydrogen atoms for each carbon atom, is known to contain distinct CH4 molecules. The formula of a compoundsuch as CH4indicates the types of atoms present, with subscripts representing the relative numbers of atoms. Water, which is a chemical compound of hydrogen and oxygen in the ratio two hydrogen atoms for every oxygen atom, contains H2O molecules. Sodium chloride is a chemical compound formed from sodium (Na) and chlorine (Cl) in a 1:1 ratio. Although the formula for sodium chloride is NaCl, the compound does not contain actual NaCl molecules. Rather, it contains equal numbers of sodium ions with a charge of positive one (Na+) and chloride ions with a charge of negative one (Cl).The substances mentioned above exemplify the two basic types of chemical compounds: molecular (covalent) and ionic. Sodium chloride, on the other hand, contains ions so we can say that it is an ionic compound.”

AcylBromide ≡ AcylHalide ⊓ ∃hasPart.AcylBromideGroup
AcylChloride ≡ AcylHalide ⊓ ∃hasPart.AcylChlorideGroup
AcylCompound ≡ OrganicCompound ⊓ ∃hasPart.AcylGroup
AcylFluoride ≡ AcylHalide ⊓ ∃hasPart.AcylFluorideGroup
AcylHalide ≡ OrganicCompound ⊓ ∃hasPart.AcylHalideGroup
AcylIodide ≡ AcylHalide ⊓ ∃hasPart.AcylIodideGroup
Alcohol ≡ OrganicCompound ⊓ ∃hasPart.HydroxylGroup
Aldehyde ≡ OrganicCompound ⊓ ∃hasPart.AldehydeGroup
Amide ≡ OrganicCompound ⊓ ∃hasPart.AmideGroup
Amine ≡ OrganicCompound ⊓ ∃hasPart.AmineGroup
Atom ⊑ ¬OrganicCompound
CarbonylCompound ≡ OrganicCompound ⊓ ∃hasPart.CarbonylGroup
CarboxylicAcid ≡ OrganicCompound ⊓ ∃hasPart.CarboxylicAcidGroup
Ester ≡ OrganicCompound ⊓ ∃hasPart.EsterGroup
Ether ≡ OrganicCompound ⊓ ∃hasPart.EtherGroup
HalogenCompound ≡ OrganicCompound ⊓ ∃hasPart.HalogenAtom
Hydrocarbon ≡ OrganicCompound ⊓ ∃hasPart.CarbonAtom ⊓ ∃ hasPart HydrogenAtom ⊓ ∀ hasPart (CarbonAtom ⊔ HydrogenAtom)
Imine ≡ OrganicCompound ⊓ ∃hasPart.ImineGroup
Ketone ≡ OrganicCompound ⊓ ∃hasPart.KetoneGroup
OrganicCompound ≡ Compound ⊓ ∃hasPart.CarbonGroup
OrganicCompound ⊑ ¬Atom
OrganicSulfurCompound ≡ OrganicCompound ⊓ ∃hasPart.OrganicSulfurG
PrimaryAmine ≡ OrganicCompound ⊓ ∃hasPart.PrimaryAmineGroup
SecondaryAmine ≡ OrganicCompound ⊓ ∃hasPart.SecondaryAmineGroup
TertiaryAmine ≡ OrganicCompound ⊓ ∃hasPart.TertiaryAmineGroup

Fig. 3. Part of the chemical ontology.

We run a use case scenario based on the essay in example 5. Firstly, we load the domain knowledge, that is the ontology in chemical domain shown in Fig. 3. Secondly, we generate the list of hypothesis based on the ontology, like in Fig.2. Thirdly,

TABLE II
SAMPLE OF TEXTUAL ENTAILMENT RESULTS.

Hypothesis	*Confidence*
Every AcylBromide is an AcylHalide that hasPart an AcylBromideGroup	0.9999997639035405
Every Alcohol is an OrganicCompound that hasPart a HydroxylGroup	0.9999999948280452
Every Aldehyde is an OrganicCompound that hasPart an AldehydeGroup	0.9999997639035405
Every Amide is an OrganicCompound that hasPart an AmideGroup	0.9999997639035405
Every OrganicSulfurCompound is an OrganicCompound that hasPart an OrganicSulfurGroup	0.9999997639035405
No Atom is an OrganicCompound	0.9999991153658677

TABLE III
GRADING 10 ESSAYS WITH 10 HYPOTHESIS (H_{10}) AND 20 HYPOTHESIS (H_{10}).

Essay	$Grade(H_{10})$	$Time(H_{10})$	$Grade(H_{20})$	$Time(H_{20})$
1	8.2	2.5	8.5	4
2	8.5	1.9	9	3.8
3	8	2.75	8.5	4.5
4	7	2.8	8	4.5
5	7.5	2.25	8	3.8
6	7.3	2	7.8	3.75
7	7.5	2.5	8	4
8	8	2.9	8.3	3.5
9	8	2.3	8	3.75
10	7.5	2.5	7.8	4

TABLE IV
COMPARISON WITH SIMILAR SYSTEMS AND HUMAN EVALUATION.

Essay ID	Our system	PaperRater	Manual evaluation
1	8.5	8.5	9
2	9	8.8	9.5
3	8	8.1	8.5
4	8.3	9	9.5
5	8	8.9	9
6	8.5	8.3	8
7	8	8.1	7.5
8	8.5	8.2	8
9	8	8.3	9
10	7.5	7.8	7

we select a specific number of hypothesis, which will be used by the component of textual entailment. Finnaly, we load the essay, and we run the component for textual entailment. The system runs each hypothesis on the essay and returns a confidence value in [0..10] for each hypothesis. Performing these steps in the essay in example 5 on six hypothesis, we obtain the results in Table II. These confidece values are used to compute the grade.

Consider a set of 10 essay to assess in the chemical domain. Assume that the domain ontology have been already generated by TextToOnto or available from various ontology repositories. The professor has the following four tasks:

1) Load the domain ontology (i.e. organic_compound.owl);
2) Generate all the hypothesis in natural language from that ontology (based on OWLNatural);
3) Select the hypothesis against which the essay should be verified;
4) Load the essay to be check if entails the selected hypothesis (based on textual entailment).

Table III shows the assessment when the user selects different number of hypothesis. The execution time varies between 1.5 and 3 minutes for 10 hypothesis, and between 3.5 and 4.5 minutes in case of 20 hypothesis. Table IV shows the results obtained after the comparison with the similar system PaperRater, and manual evaluation.

IV. DISCUSSION AND RELATED WORK

The current methods of essay scoring can be categorized into two classes: holistic scoring and rubric-based scoring. In holistic scoring, the essay is assessed and a single score selected from a predefined score range is assigned as an overall score [24]. In the analytical or rubric-based scoring method, essays are assessed on the basis of a certain set of well-defined features [24]. Each feature has a scale associated with it and the final score awarded to the essay is the sum of scores of all the essay rubrics/features.

The National Assessment Program Literacy and Numeracy (NAPLAN) [7] rubric of persuasive essay grading lists a set of criteria for marking persuasive writing. The Spelling Mark algorithm is developed to formalise the NAPLAN rubric for marking spelling based on common heuristics and rules of the English language. The first step is to obtain the total number of words in the essay and the number of spelling errors in the essay. Then, each word is categorized based on the difficulty level into one of four classes: simple, common, difficult or challenging, and number of correct and incorrect words in each category is counted. The final step in the algorithm is to assign the spelling mark according to the set of rules [7].

PaperRater (https://www.paperrater.com/) is an automated proofreading tool that combines natural language processing, machine learning, information retrieval and data mining to help students write better. PaperRater is used by schools and universities in over 46 countries to help students improve their writing and check for plagiarism. The system has a core NLP engine using statistical and rules to extract language features from essays and robustly translate that into statistical models. The three major features offered by the program are: spell checker, grammar checker, and plagiarism checker. The system has also a vocabulary builder tool designed to help students learn proper usage of more sophisticated words.

A complementary line of research is given by argumentative writing support systems [13]. Assisting students in essays with structured and sound arguments is an important educational goal [19]. Hence, various argumentative support systems has been applied in collaborative educational environments [2], [17]. Persuasive essays are extensively studied in the context of automated essay grading. Since argument structure is crucial for evaluating essay quality, [21] identifies the argumentative discourse structure by means of discourse marking. The goal is to model argument components as well as argumentative relations that constitute the argumentative discourse structure in persuasive essays. The annotation scheme includes three argument components (major claim, claim and premise) and two argumentative relations (support and attack) [21]. The legal educational system LARGO [16] uses an ontology containing the concepts test, hypothetical, and fact situation and roles such as distinction and modification. The system Convince Me [18] employs more scientific-focused primitives such as hypothesis and data elements with explain and contradict links. Other systems such as Rationale [22] provide more expansive primitive sets that allow users to construct arguments in different domains. Our work fits in this context by using natural language processing for argument mining. The automatic grading mechanism could benefit from a system able to identify arguments in a student essay.

V. Conclusion

We developed a NLP tool for automatically essay grading in different domains. We enacted textual entailment to compare the text written by a student with the requirments of a human evaluator. These requirments are generated in natural language from a given domain ontology. The main benefit is that the user can select different ontologies for processing text in various domains. However, the confidence in the assessment depends on the precision of the textual entailment method that relies on domain datasets for training.

In line with [15], we currently aim to assess the confidence in the system by performong more comparisons with humans that evaluate essays.

Acknowledgments

We thank the reviewers for valuable comments. This research was supported by the Technical University of Cluj-Napoca, Romania, through the internal research project GREEN-VANETS.

References

[1] I. Androutsopoulos and P. Malakasiotis, “A survey of paraphrasing and textual entailment methods,” *Journal of Artificial Intelligence Research*, pp. 135–187, 2010.

[2] F. Belgiorno, R. De Chiara, I. Manno, M. Overdijk, V. Scarano, and W. van Diggelen, “Face to face cooperation with coffee,” in *Times of Convergence. Technologies Across Learning Contexts*. Springer, 2008, pp. 49–57.

[3] P. Cimiano and J. Völker, “Text2Onto,” in *Natural language processing and information systems*. Springer, 2005, pp. 227–238.

[4] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, “Gate: an architecture for development of robust hlt applications,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 168–175.

[5] I. Dagan, O. Glickman, and B. Magnini, “The pascal recognising textual entailment challenge,” in *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer, 2006, pp. 177–190.

[6] A. Fader, S. Soderland, and O. Etzioni, “Identifying relations for open information extraction,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

[7] A. Fazal, F. K. Hussain, and T. S. Dillon, “An innovative approach for automatically grading spelling in essays using rubric-based scoring,” *Journal of Computer and System Sciences*, vol. 79, no. 7, pp. 1040–1056, 2013.

[8] N. E. Fuchs, K. Kaljurand, and T. Kuhn, “Attempto controlled english for knowledge representation,” in *Reasoning Web*. Springer, 2008, pp. 104–124.

[9] D. Galanis and I. Androutsopoulos, “Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system,” in *Proceedings of the Eleventh European Workshop on Natural Language Generation*. Association for Computational Linguistics, 2007, pp. 143–146.

[10] A. C. Graesser, D. S. McNamara, and J. M. Kulikowich, “Coh-Metrix providing multilevel analyses of text characteristics,” *Educational Researcher*, vol. 40, no. 5, pp. 223–234, 2011.

[11] A. C. Graesser, D. S. McNamara, M. M. Louwerse, and Z. Cai, “Coh-metrix: Analysis of text on cohesion and language,” *Behavior research methods, instruments, & computers*, vol. 36, no. 2, pp. 193–202, 2004.

[12] K. B. H. Cunningham, D. Maynard and V. Tablan, “GATE: A framework and graphical development environment for robust NLP tools and applications,” in *40th Annual Meeting of the ACL*, 2002.

[13] I. A. Letia and A. Groza, “Arguing with justifications between collaborating agents,” in *Argumentation in Multi-Agent Systems*. Springer Berlin Heidelberg, 2012, pp. 102–116.

[14] B. Magnini, R. Zanoli, I. Dagan, K. Eichler, G. Neumann, T.-G. Noh, S. Pado, A. Stern, and O. Levy, “The excitement open platform for textual inferences,” *ACL 2014*, p. 43, 2014.

[15] M. R. Manap, “Comparison between web-based automated essay scoring software and human esl essay assessment: A preliminary investigation,” in *INTERNATIONAL CONFERENCE ON SOCIAL SCIENCES & HUMANITIES 2012*, 2012.

[16] N. Pinkwart, C. Lynch, K. Ashley, and V. Aleven, “Re-evaluating LARGO in the classroom: Are diagrams better than text for teaching argumentation skills?” in *Intelligent Tutoring Systems*. Springer, 2008, pp. 90–100.

[17] N. Pinkwart and B. M. McLaren, *Educational technologies for teaching argumentation skills*. Bentham Science Publishers, 2012.

[18] P. Schank and M. Ranney, “Improved reasoning with Convince Me,” in *Conference companion on Human factors in computing systems*. ACM, 1995, pp. 276–277.

[19] O. Scheuer, F. Loll, N. Pinkwart, and B. M. McLaren, “Computer-supported argumentation: A review of the state of the art,” *International Journal of Computer-Supported Collaborative Learning*, vol. 5, no. 1, pp. 43–102, 2010.

[20] M. D. Shermis and J. Burstein, *Handbook of automated essay evaluation: Current applications and new directions*. Routledge, 2013.

[21] C. Stab and I. Gurevych, “Identifying argumentative discourse structures in persuasive essays,” in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)(Oct. 2014), Association for Computational Linguistics, p.(to appear)*.

[22] T. Van Gelder, “The rationale for rationale,” *Law, probability and risk*, vol. 6, no. 1-4, pp. 23–42, 2007.

[23] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, “Textrunner: open information extraction on the web,” in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2007, pp. 25–26.

[24] R. K. Y.W. Lee, C. Gentile, *Toward automated multi-trait scoring of essays: Investigating links among holistic, analytic, and text feature scores*. Appl. Linguist., 2010.