Identification of Sentence-to-Sentence Relations using a Textual Entailer

We show in this article how an approach developed for the task of recognizing textual entailment relations can be extended to identify paraphrase and elaboration relations. Entailment is a unidirectional relation between two sentences in which one sentence logically infers the other. There seems to be a close relation between entailment and two other sentence-to-sentence relations: elaboration and paraphrase. This close relation is discussed to theoretically justify the newly derived approaches. The proposed approaches use lexical, syntactic, and shallow negation handling. When compared to other paraphrase and elaboration approaches it produces similar or better results. The proposed approach also offers significantly better results than several baselines. We report results on several data sets: the Microsoft Research Paraphrase corpus, a benchmark for evaluating approaches to paraphrase identification, and a data set collected from high-school students' interactions with an intelligent tutoring system iSTART, which includes both paraphrase and elaboration utterances.

1. Introduction

There are many applications in which it is useful to identify semantic relationships between two sentences. One such relationship is paraphrase, a text-to-text relation between two non-identical text fragments that express the same meaning in different ways. We focus on sentential paraphrases in this article as opposed to paraphrases among larger (paragraphs) or smaller text

chunks (phrases or clauses). As an example of a paraphrase, we show the two sentences below [from the Microsoft Research Paraphrase Corpus (Dolan, Quirk, & Brockett 2004)], where Text B is a paraphrase of Text A and vice versa.

Text A: *A strong geomagnetic storm was expected to hit Earth today with the potential to affect electrical grids and satellite communications.*

Text B: *A strong geomagnetic storm is expected to hit Earth sometime today and could knock out electrical grids and satellite communications.*

Paraphrases are important in a number of applications. In Natural Language Generation, paraphrases are a method to increase diversity of generated text (Iordanskaja, Kittredge, & Polgere 1991). Paraphrases are useful in Intelligent Tutoring Systems with natural language interaction (ITSs; Graesser *et al.* 2005; McNamara *et al.* 2007) in which the student-system interaction is driven primarily by a dialogue in natural language. In such ITSs, it is important to assess, for instance, whether student articulated answers to deep questions, e.g. conceptual physics questions, are similar-to/paraphrases-of ideal answers. In Question Answering, multiple answers that are paraphrases of each other could be considered as evidence for the correctness of the answer (Ibrahim, Katz, & Lin 2003).

The paraphrase relation between two texts is closely related to the entailment relation. Textual entailment is the task of deciding, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text (RTE, Recognizing Textual Entailment

challenge; Dagan, Glickman, & Magnini 2005). We say that T - the entailing text, entails H - the entailed hypothesis. For instance, T entails H in the example below.

T: *Seeing the huge market potential, Yahoo bought Overture Services, Inc.*
H: *Yahoo took over Overture.*

The task of recognizing textual entailment is relevant to a large number of applications, including Machine Translation, Question Answering, and Information Retrieval (Dagan, Glickman, & Magnini, 2005). A paraphrase can be looked at as a bidirectional entailment relation between the two text fragments. In other words, text A is a paraphrase of text B if and only if A entails B and B entails A. Therefore, having a solution to the unidirectional problem of entailment would allow us to extend it to address the paraphrase identification problem.

While the paraphrase relation between two sentences is one of semantic equivalence, an elaboration indicates that one of the sentences *elaborates*, i.e., adds extra information to, the other sentence. The elaboration relation between sentences has an interesting twist: it can be regarded as a reverse entailment. That is, the elaborated sentence is usually longer and content richer than the benchmark sentence thus entailing/subsuming the benchmark. As an example of an elaboration relation between two sentences, we show a sentence from a science textbook and an explanation of it as produced by a student when prompted to self-explain it. The student self-explanations in this study are all reproduced as typed by the student.

Text: *These amino acids are hooked together to make proteins at very small organelles called ribosomes.*

Elaboration: *the amino acids which is a chemical are hooked to nucleolus to make a protein called ribosomes thats very small.*

This is a real example from iSTART (McNamara, Boonthum, et al., 2007), an intelligent tutoring system that teaches students reading strategies. Students must *self-explain* a sentence from a previously read science text. The student must self-explain the sentence (or *benchmark*) using one of several reading strategies: *paraphrasing* the text, making *bridging inferences* between the current sentence and prior text, and *elaborating* the text with links to what the reader already knows.

This article describes a fully-implemented software system that embeds newly proposed approaches to paraphrase and elaboration identification. The paraphrase identification approach is based on a previously proposed approach to entailment proposed by HIDDEN REFERENCE (YEAR) that relies on lexical, syntactic, and shallow negation handling. The entailment approach offered competitive results as compared to other approaches to entailment (Dagan, Glickman, & Magnini, 2004-2005). The entailment approach (HIDDEN REFERENCE, YEAR) uses lexical, syntactic, synonymy and antonymy (for negation handling) information. The synonymy and antonymy information is extracted from a thesaurus, i.e., WordNet (Miller, 1995). No deeper processing, world knowledge, or automated reasoning is used due to the high costs and scalability issues. Using an approach such as the Entailer, we are able to answer the question of how well combined lexical, syntactic, and negation information can solve the tasks of recognizing entailment, paraphrase, and elaboration. The entailment approach was successfully tested on data from the task of recognizing textual entailment (RTE; Dagan, Glickman, & Magnini 2004-2005). In our approach to entailment, we first map each T-H pair onto two graphs,

one for H and one for T, with nodes representing main concepts and links indicating syntactic dependencies among concepts as encoded in H and T, respectively. An entailment score, *entscore*(*T;H*) (see Equation 1), is then computed quantifying the degree to which the T-graph subsumes the H-graph. The score is so defined to be non-symmetric, i.e., *entscore*(*T;H*) ≠ *entscore*(*H; T*). We show in this article how to extend this approach to handle paraphrases.

The rest of the article is organized as follows. The *Motivation* section presents the advantages of developing algorithms to recognize paraphrase and elaboration relations. In particular, we show their value in educational systems such as iSTART. The next section, *Related Work*, presents previous research on entailment, elaboration, and paraphrase identification that are closely related to our work. The *Approach* section presents in detail our lexico-syntactic approach to paraphrase and elaboration identification. Following this, the *Experiments and Results* describes experiments on the standard Microsoft Research Paraphrase Corpus and on data from the intelligent tutoring system iSTART. The *Discussion* section analyzes the results obtained, while *Further Work* presents ` plans for the future. A *Conclusions* section ends the article.

2. Motivation

Paraphrase and elaboration identification is of particular value to educational systems. A major challenge for Intelligent Tutoring Systems (ITSs) that incorporate natural language interaction is to accurately evaluate users' contributions and to produce appropriate feedback. Available research in the learning sciences indicates that guided feedback and explanation is more effective than simply providing an indication of *rightness* or *wrongness* of student input (Mark & Greer, 1995; McKendree, 1990;Azevedo & Bernard, 1995).

The ITS in this study, iSTART, uses pedagogical agents to provide young adolescent to college-aged students with tutored self-explanation and reading strategy training. iSTART is designed to improve students' ability to self-explain by teaching them to use reading strategies such as *elaboration*, *bridging*, and *paraphrasing*. For example, paraphrasing requires students to restate sentences in their own words; such a process helps students to monitor their comprehension and also activate knowledge relevant to the target information. Following *introduction* and *practice* phases of the iSTART training, the final practice phase has students use reading strategies by typing *self-explanations* of sentences from science texts. For example, the following sentence, called Text (T), is from a science textbook and the student input, called self-explanation (SE), is reproduced from a recent iSTART experiment.

T: *The largest and most visible organelle in a eukaryotic cell is the nucleus.*

SE: *the nucleusis the center of the cell it contains the ribsome and more.*

The object of existing iSTART algorithms is to assess which strategy (or, *type* of self-explanation) has been attempted by the student. However, further algorithms are needed to assess how close in meaning the self-explanation is to the target sentence (i.e., is the self-explanation a *paraphrase* of the target sentence? Is it an *elaboration*? Or is it *entailed* by the target sentence?). Thus, the more accurately the self-explanations can be assessed, the more appropriately the system can provide feedback to the user. In this study, we explore these evaluations of self-explanations using extensions of an approach for the task of recognizing textual entailment. Additionally, we compare the approaches to a variety of textual-assessment metrics.

3. Related Work

Our work is related to efforts in the areas of paraphrase identification, textual entailment, and assessing textual relatedness in intelligent tutoring systems. To better position our work, we briefly describe next such previous efforts.

Paraphrase identification is the task of deciding whether two text fragments, usually sentences, are paraphrases of each other. Our approach should not be confused with paraphrase extraction which is the task of collecting pairs of text fragments that are paraphrases of each other from different sources. Paraphrase extraction research uses parallel translations of same source text from a foreign language (Barzilay & McKeown 2001), named entities anchors in candidate sentences, and sequence alignment algorithms to extract paraphrases. Different translations of the same original sentence guarantees the paraphrase relation among the corresponding sentences in the parallel translations. Paraphrase extraction from the web was also successfully attempted (Dolan, Quirk, & Brockett 2004).

Paraphrase identification has been previously explored, most notably by Kozareva and Montoyo (2006), Mihalcea, Corley, and Strapparava (2006), and Qiu, Kan, and Chua (2006), who, like us, all used the same standard data set (the Microsoft Research Paraphrase Corpus) to evaluate their methods. A direct comparison with their methods is thus possible. Kozareva and Montoyo (2006) proposed a machine learning approach based on lexical and semantic information (e.g., a word similarity measure based on WordNet). They model the problem of paraphrasing as a classification task. Their model uses a set of linguistic attributes and three different machine learning algorithms (Support Vector Machines [SVM], k-Nearest Neighbors, and Maximum Entropy) to induce classifiers. The classifiers are built in a supervised manner from labeled training data in the Microsoft Research Paraphrase Corpus. All of the attributes that

they defined are bidirectional, i.e., they capture sentence similarity in both directions. Three types of experiments were conducted. In one experiment, they simply compared the three types of classifiers; in a second experiment they tried different attribute mixtures from their original set of attributes; while in a third experiment they combined several classifiers. The SVM classifier outperformed the others in all experiments. Mihalcea, Corley, and Strapparava (2006) introduced several *simple* approaches that rely on corpus-based and knowledge-based word-to-word similarity measures. The measures rely on path or node distance in WordNet, e.g. Leacock and Chodorow (1998), and statistical distributions of words in large corpora [Latent Semantic Analysis, (Landauer et al, 2007)]. Further, they proposed a so-called *combined* approach that computes a simple average among the output of the simple approaches.

The last related work that we can directly compare ours to is Qiu, Kan, and Chua (2006). They present a two-step approach. In the first step, they identify similarities between two sentences in a possible paraphrase relation. In a second step, dissimilarities among the sentences are detected using machine learning methods. The core idea of their approach is to use predicate argument *tuples* that capture both lexical and syntactic dependencies among words to find similarities between sentences. Our method, which relies on an approach proposed by HIDDEN REFERENCE (YEAR), differs from the methods of Kozareva and Montoyo (2006) and Mihalcea, Corley, and Strapparava (2006) in several ways. First, we use syntactic information as one of the components of our approach. Second, our lexical component relies on word overlap enhanced with synonymy relations from WordNet as opposed to word-to-word similarity measures. Lastly, our approach incorporates negation handling based on antonymy relations in WordNet. With respect to the method of Qiu, Kan, and Chua (2006), our approach differs almost in the same way it differs from Kozareva and Montoyo (2006) and Mihalcea, Corley, and

Strapparava (2006) except the use of syntactic information. Qiu, Kan, and Chua (2006) use paths in syntactic trees as features in modeling dissimilarities between two sentences. In the *Experiments and Results* section, we compare the results reported by Kozareva and Montoyo (2006), Mihalcea, Corley, and Strapparava (2006), and Qiu, Kan, and Chua (2006) with results obtained with our approach.

The task of textual entailment was treated in the recent past in one form or another by research groups ranging from informational retrieval to language processing. In one of the earliest explicit treatments of entailment, Monz & de Rijke (2001 proposed a weighted bag of words approach to entailment. More recently, Dagan and Glickman (2004) presented a probabilistic approach to textual entailment based on lexico-syntactic structures. Pazienza, Pennacchiotti, and Zanzotto (2005) used a syntactic graph distance approach for the task of textual entailment. More recently still, Kouylekov and Magnini (2005) approached the entailment task with a tree edit distance algorithm on dependency trees. One distinct feature of our lexico-syntactic approach is the negation handling. None of the above mentioned approaches addressed negation.

In intelligent tutoring systems, text relatedness metrics such as Latent Semantic Analysis (LSA; Landauer et al., 2007) and overlap-indices have proven to be extremely effective measures for a great variety of the systems that analyze natural language and discourse, such as Coh-Metrix (Graesser, et al., 2004), iSTART (McNamara et al., 2004), and AutoTutor (Graesser et al., 2005). Despite such successes, the need remains for new measures of textual assessment to augment existing measures and thereby better assess textual comparisons. We assess in this article a variety of more established textual relatedness assessment metrics (e.g., LSA and Content-Overlap), and compare them to newer approaches such as the Entailer (HIDDEN

REFERENCE, YEAR) and *MED* (McCarthy et al., 2007). Each of these measures provides a unique approach to assessing the relatedness between text fragments. Therefore, we believe that a combination of such approaches (or soft constraints) is likely to offer the user the fullest range of feedback. We offer an overview of each of the approaches later. The comparison is performed on the data collected from live student-iSTART interactions.

4. Approach

Our solution for paraphrase identification is an extension of an approach to the unidirectional relation of entailment. The approach to recognizing textual entailment is based on the idea of *subsumption*. In general, an object X subsumes an object Y if X is more general than or identical to Y, or alternatively we say Y is more specific than X. The same idea applies to more complex objects, such as structures of interrelated objects. Applied to textual entailment, subsumption translates into the following: hypothesis H is entailed from T if and only if T subsumes H. The solution has two phases: (I) map both T and H into graph structures and (II) perform a subsumption operation between the T-graph and H-graph. The processing flow is shown in Figure 1. The approach takes as input two raw text fragments and returns a decision (TRUE or FALSE) indicating whether T entails H.

Phase I: From Text to Graph Representations. The two text fragments involved in a textual entailment decision are initially mapped onto a graph representation. The graph representation we employ is based on the dependency graph formalism of Mel'cuk (1998). The mapping process has three stages: preprocessing, dependency graph generation, and final graph generation. In the preprocessing stage, the system separates the punctuation from words (tokenization), maps morphological variations of words to their base or root form

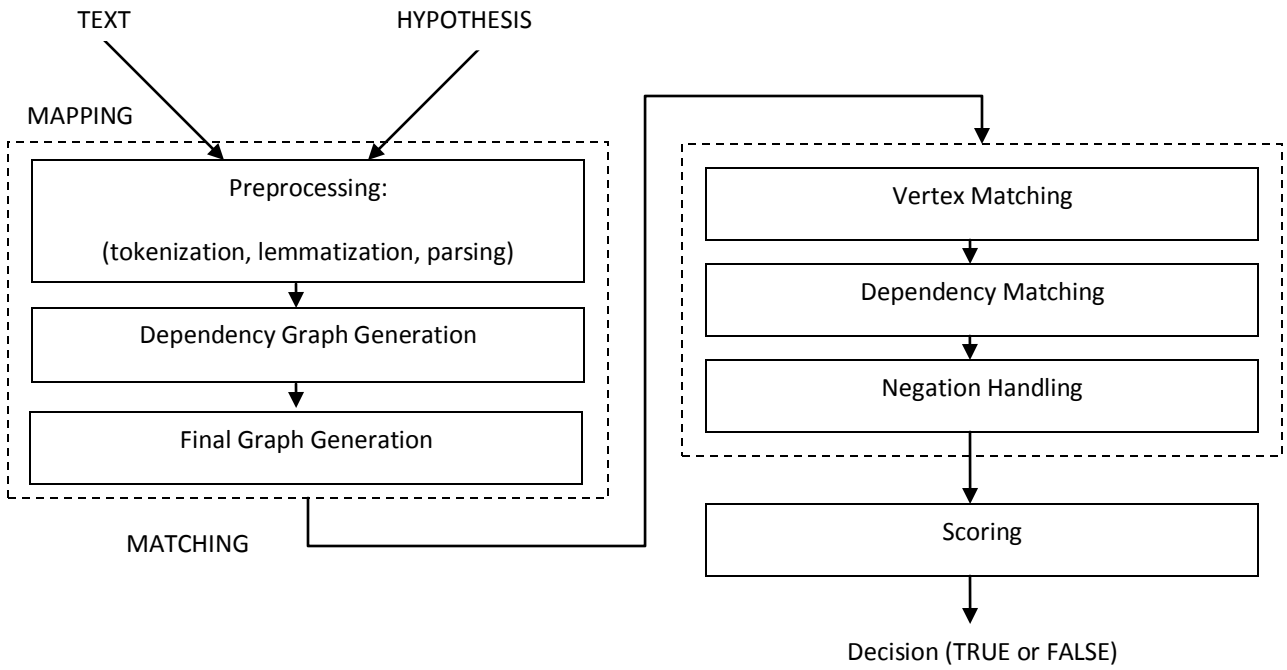

Figure 1: Processing flow in Entailer.

(lemmatization), assigns part-of-speech labels to each word (tagging), and assesses the inter-relationship of major phrases within the texts (parsing). Additional preprocessing operations are also performed, such as detecting collocations of common nouns [for more details see (HIDDEN REFERENCE, YEAR)]. For instance, *joint venture* is identified as a collocation and thus a single node in the graph is generated for it instead of two, i.e., one for *joint* and one for *venture*. The second stage (dependency graph generation) is the actual mapping from text to the graph representation. This

mapping is based on information from parse trees generated during the parsing process. A parse tree groups words in a sentence into phrases and organizes these phrases into hierarchical tree structures from which we can easily detect syntactic dependencies among concepts. The system uses Charniak's (2000) parser to obtain parse trees and Magerman's (1994) head-detection rules to obtain the head of each phrase. A dependency tree is generated by linking the head of each phrase to its modifiers in a systematic mapping process. The parser is also used for

part-of-speech tagging, i.e., no separate part-of-speech tagger is used in preprocessing. In the third stage (final graph generation), the dependency tree generated in the second stage is refined. The dependency tree encodes exclusively local dependencies (head-modifiers), as opposed to remote dependencies, such as the remote subject relation between *bombers* and *enter* in *The bombers had not managed to enter the embassy compounds* (from RTE, Dagan, Glickman, & Magnini 2004-2005). Thus, in this stage, the dependency tree is transformed onto a dependency graph by generating remote dependencies and direct relations between content words. Remote dependencies are computed by a naive-Bayes functional tagger (HIDDEN REFERENCE, YEAR). Direct relations are generated using a simple procedure that directly links content words by eliminating certain types of intermediate relations, e.g. *mod*. For an example of a direct relation between content words, consider the sentence *I visited the manager of the company*. For this sentence, the modifier *(mod)* dependency between the noun *manager* and its attached preposition *of* is replaced with a direct relation between the prepositional head *manager* and prepositional object *company*. Once graph representations are obtained, a graph subsumption operation is initialized, as described below.

Phase II: Graph Subsumption. The textual entailment problem is mapped into a specific example of graph isomorphism called subsumption (also known as containment). Isomorphism in graph theory addresses the problem of testing whether two graphs are the same. A graph G = (V, E) consists of a set of nodes or vertices V and a set of edges E. Graphs can be used to model the linguistic information embedded in a sentence: vertices represent concepts (e.g., *bombers*, *joint venture*) and edges represent syntactic relations among concepts (e.g., the edge labeled *subj* connects the verb *managed* to its subject *bombers*). When using such graph

representations, the Text (T) entails the Hypothesis (H) if and only if the hypothesis graph is subsumed (or contained) by the text graph.

The subsumption algorithm for textual entailment has three major steps: (1) find an isomorphism between V_H (set of vertices of the Hypothesis graph) and V_T (set of vertices of the Text graph); (2) check whether the labeled edges in H, E_H, have correspondents in E_T ; (3) compute score. Step 1 is more than a simple word-matching method because if a vertex in H does not have a direct correspondent in T, a thesaurus is used to find all possible synonyms for vertices in T. In addition, vertices in H have different priorities, such as the fact that head words are more important than modifiers. Step 2 takes each relation in H and checks its presence in T. The checking is augmented with relation equivalences among linguistic phenomena such as possessives, and linking verbs (e.g. *be*, *have*). For instance, *tall man* would be equivalent to *man is tall*. Lastly, a normalized score for vertices and edge mapping is computed. The score for the entire entailment is the weighted sum of each individual vertex and edge matching score (see Equation 1). The weights for the vertex and edge matching scores are given by the parameters α and β, respectively. γ is the free term which can be used to bias the score towards making more optimistic decisions (positive values for γ) or not (negative values for γ).

$$entscore(T,H) = (\alpha \times \frac{\sum_{V_h \in H_v} \max_{V_t \in T_v} match(V_h, V_t)}{|V_h|} + \beta \times \frac{\sum_{E_h \in H_e} \max_{E_t \in T_e} match(E_h, E_t)}{|E_h|} + \gamma) \times (\frac{1+(-1)^{\#neg\_rel}}{2})$$

Equation 1. Entailer Score is a weighted sum of lexical and syntactic matching. The overall score has a term that accounts for negation.

Negation. We also handle two broad types of negation: explicit and implicit. Explicit negation is indicated by particles such as: *no, not, neither ... nor* and their shortened forms *n't*. Implicit negation is present in text via deeper lexico-semantic relations among different linguistic

expressions; the most obvious example is the *antonymy* relation among words, which can be retrieved from WordNet (Miller, 1995). Negation is regarded as a feature of both Text and Hypothesis and it is accounted for in the score after the entailment decision for the Text-Hypothesis pair ignoring negation is made. If one of the text fragments is negated the decision is reversed while if both are negated the decision is retained (double-negation), and so forth. For example, the Text *Yahoo bought Overture* does not entail Hypothesis *Yahoo did not buy Overture* because even though Text subsumes the Hypothesis ignoring negation (*Yahoo did buy Overture*), the presence of negation reverses that decision. In Equation 1 the term *#neg rel* represents the number of negation relations between T and H. We have plans to further improve the basic negation handling algorithm in the future. One simple extension to the current algorithm will be to assess other plausible negation concepts such as the words *denied, denies, without, ruled out* (Chapman et al., 2001). As a second extension, we will address negation scope for verbs. For instance, in the sentence *The bombers had not managed to enter the embassy compounds* the negation of the verb *manage* extends to its subordinate verb *enter*.

Elaboration – Reverse Entailment. The entailment approach between a Text and Hypothesis generates a score that indicates the degree of the Hypothesis being subsumed by the Text. In RTE challenges (Dagan, Glickman, & Magnini 2004-2005), the Text is longer, word-wise, than the Hypothesis, which is typical for entailment relations among two sentence pairs. However, there are other sentence-to-sentence relations, such as elaboration, in which the original sentence, i.e., Text, is shorter than its elaborated counterpart, i.e., Hypothesis. In an elaboration identification task, the challenge would be to decide whether the Hypothesis is an elaboration of the Text. In such cases, the entailment approach can still be applied in a reverse manner because the Hypothesis is longer than the Text in an elaboration relation. Thus, we must

check whether the Hypothesis graph subsumes the Text graph. We could simply compute the score entscore (H, T) where the roles of the Text and Hypothesis in the regular entailment case are reversed. This score will be identified in the *Experiments and Results* section as Ent-Rev to avoid confusion with the scores of human judgments on the elaboration dimension. The entailment approach and its associated score will be identified later as Ent-For (from *Forward*).

$$elaboration(T, H) = entscore(H, T)$$

Paraphrase - Bidirectional Entailment. The approach to entailment can be extended to handle paraphrases. The idea is based on the observation that text A is a paraphrase of text B if and only if text A entails text B and text B entails text A. Thus, a paraphrase score that is the average of entscore(A, B) and entscore(B, A) can be defined. This score will be identified in the *Experiments and Results* section as Ent-Avg to avoid confusion with the scores of human judgments on the paraphrase dimension.

$$paraphrase(A, B) = \frac{entscore(A, B) + entscore(B, A)}{2}$$

5. Experiments and Results

In this section, we present the details of the experiments we conducted to evaluate the performance of the newly proposed approach to paraphrase and elaboration identification.

Microsoft Research Paraphrase Corpus. The Microsoft Research Paraphrase corpus (MSRP) is a standard corpus for paraphrase identification (Dolan, Quirk, & Brockett, 2004). MSRP corpus (Dolan, Quirk, & Brockett 2004) contains 5801 pairs of sentences collected from various news sources on the web. Each pair is accompanied by "a judgment reflecting whether multiple human annotators considered the two sentences to be close enough in meaning to be considered close paraphrases." The corpus is divided into two subsets: *training* and *test*. There

are 2753 TRUE and 1323 FALSE paraphrase instances in the training subset. The test data subset contains 1147 TRUE instances and 578 FALSE instances. For evaluation purposes, we also generated balanced data sets of 1000-1000 (TRUE-FALSE split) and 500-500 instances for training and testing purposes, respectively.

The results reported here are on the test data sets while the system development/tuning was conducted on the training set. The evaluation is automatic and follows the guidelines from RTE (Dagan, Glickman, & Magnini 2004, 2005). The judgments (classifications) returned by the system are compared to those manually assigned by the human annotators (i.e., the gold standard). The percentage of matching judgments provides the *accuracy* of the run, which is the fraction of correct responses.

As a second measure, a Confidence-Weighted Score [CWS, also known as average precision; (Dagan, Glickman, & Magnini 2004, 2005)] is computed. Judgments of the test examples are sorted by their confidence (in decreasing order from the most certain to the least certain), calculating the following measure:

$$\frac{1}{n} = \sum_{i=1}^{n} \frac{\#\text{–}correct - up - to - pair - i}{i}$$

where *n* is the number of the pairs in the test set, and *i* ranges over the pairs. CWS varies from 0 to 1 (perfect score), and rewards the system's ability to assign a higher confidence score to correct judgments.

We used the training data to estimate the parameters of the score equation and then applied the equation with the best found parameters to the test data. We used linear regression to

estimate the values of the parameters and also experimented with balanced weighting ($\alpha = \beta = 0.5$, $\gamma = 0$). The balanced weighted scheme provides better results. The performance figures reported below are obtained with this balanced scheme. The score provided by the formula in Equation 1 is further used to find the paraphrase decision (TRUE or FALSE) and the level of confidence. Depending on the value of the overall score, different levels of confidence are assigned. For instance, an overall score of 0 leads to FALSE paraphrase with a maximum confidence of 1. In summary, we obtained an accuracy of 0.7061 and a CWS score of 0.8068 on the standard data set (see Ent-*Avg* row in Table 1). The results produced by our approach on the test data are significant ($p < 0.01$) when compared to the uniform baseline of always guessing the most frequent class (the TRUE class in the MS Paraphrase Corpus) and to the *random baseline* of randomly guessing TRUE or FALSE with equal probability. Our high CWS score indicates that our system is a *confident* paraphrase identifier. This argument becomes stronger when one thinks of the limited array of resources we use: lexical information enhanced with synonymy from WordNet, syntactic information in the form of dependencies, and antonymy information from WordNet to handle negation. By comparison, other approaches use deeper representations, which are more expensive to build and use, heavier resources to make paraphrase and entailment decisions (Dagan, Glickman, & Magnini 2004-2005).

Intelligent Tutoring Systems Experiments. The evaluation of the proposed approaches to paraphrasing and elaboration for assessing users' textual input in ITS environments focuses on data from the ITS, iSTART (Interactive Strategy Training for Active Reading and Thinking; McNamara, Levinstein, & Boonthum, 2004). Not only do we evaluate the proposed approaches but we compare them to a variety of more established textual relatedness assessment metrics (e.g., LSA and Content-Overlap), and to newer approaches such as *MED* (McCarthy et al.,

2007). Each of these measures provides a unique approach to assessing the relatedness between text fragments. While the best approach could eventually be used to assess student input based on which feedback is provided, we believe that a combination of these approaches (or soft constraints) is likely to offer the user the fullest range of feedback. Below, we offer an overview of each of the approaches used in this study.

Table 1. Performance and comparison of different approaches on MSRP corpus [* - results from (Mihalcea, Corley, & Strapparava, 2006)].

System	CWS	Accuracy	Precision	Recall	F-measure
Uniform Baseline	0.6737	0.6649	0.6649	1.0000	0.7987
Random Baseline*	-	0.5130	0.6830	0.5000	0.5780
Ent-For	0.7852	0.6788	0.7502	0.7751	0.7624
Ent-Avg	**0.8068**	0.7061	0.7207	0.9111	0.8048
Ent-Rev	0.7794	0.6736	0.7517	0.7602	0.7560
Kozareva-SVM	-	0.6986	**0.9346**	0.7066	0.8048
Mihalcea-PMI-IR	-	0.6990	0.7020	0.9520	0.8100
Mihalcea-L&C	-	0.6950	0.7240	0.8700	0.7900
Mihalcea-combined	-	0.7030	0.6960	**0.9770**	0.8130
Qiu	-	**0.7250**	0.7250	0.9340	**0.8160**

Table 2. Performance and comparison of different approach on the balanced corpus derived from the MSRP corpus.

System	CWS	Accuracy	Precision	Recall	F-measure
Uniform Baseline	0.4936	0.5000	0.5000	0.1000	0.6667
Random Baseline	0.5112	0.4880	0.4880	0.4900	0.4890
Ent-For	0.6700	0.6350	0.6060	**0.7720**	0.6790
Ent-Avg	**0.7046**	**0.6590**	**0.6470**	0.7000	0.6724
Ent-Rev	0.6732	0.6380	0.6092	0.7700	**0.6802**

Latent Semantic Analysis. LSA is a statistical technique for representing world knowledge based on large corpora of texts (Landauer et al., 2007). LSA uses a general form of factor analysis (singular value decomposition) to condense a highly dimensional representation of a very large corpus of texts to 300-500 dimensions. These dimensions represent how words (or group of words) co-occur across a range of documents within a large corpus (or space). Unlike content overlap indices, LSA affords tracking words that are semantically similar, even when they may differ morphologically.

Content Overlap. Content-overlap assesses the percentage of common nouns occuring in adjacent sentences. The more common content words two sentences have the more likely they

have the same meaning, i.e. they are paraphrases of each other. Such measures have been shown to aid in text comprehension and reading speed (Kintsch & Van Dijk, 1978).

Minimal Edit Distances (MED). *MED* (McCarthy et al., 2007) is a computational tool designed to evaluate text relatedness by assessing the similarity of lexicon in adjacent sentences. *MED* is a combination of measuring Levenstein distances (1966) and string theory matching (Dennis, 2006). Essentially, *MED* functions like a spellchecker; that is, it converts lexicon into unique character representations and then searches for the shortest route through which two such strings can be matched. The evaluations of the possible routes result in a set of *costs*: shifting the string (right or left) has a cost of 1; deleting a character costs 1; and inserting a character costs 1. Once the cost has been calculated, the value is divided by the number of elements in the string. *MED* scores are continuous, with a score of zero representing an identical match. *MED*'s major benefit over simple co-occurrence indices is that structural variation can be assessed. Thus, for *MED*, *the cat chased the dog* is different from *the dog chased the cat* (see Table 3).

Table 3. MED Evaluations for "The dog chased the cat."

	MED
The dog chased the cat.	0
The cat chased the dog.	0.267
The cats chased the dogs.	0.533
The cat didn't chase the dog.	0.941
Elephants tend to be larger than mice.	1.263

Intelligent Tutoring Systems Corpus. In order to test the textual relatedness approaches outlined above, we used a natural language corpus of iSTART user input statements. The data *pairs* used to make the corpus were generated from an iSTART experiment conducted with 90 high-school students drawn from four 9th grade Biology classes (all taught by the same teacher). Overall, the experiment generated 826 sentence pairs. The average length of the combined sentence pairs was 16.65 words (*SD* = 5.63). As an example, the following four self-explanations were made (reproduced without correction) by students responding to the target sentence *Sometimes a dark spot can be seen inside the nucleus*:

1) yes i know that can be a dartkn spot on .think aboyt what thje sentence

2) in dark spots you can see inside the nucleus and the cell

3) if you ever notice that a dark spot can be seen inside the nucleus sometime

4) the nucleus have a dark spot that sometimes be seen.its located in the inside of the nucleus.

The corpus was evaluated by hand on three different categories of textual similarity: *entailment*, *elaboration*, and *paraphrase* (see Table 4 for examples). To assess the pairs, three discourse processing experts evaluated each sentence pair on each of the three dimensions of similarity. Prior to the actual analysis, the raters were trained on a subset (100 data pairs) of the *iSTART* corpus. Each pair (for each category) was given a rating of 1 (min) to 6 (max). Training was considered successful when the *r* value for each of the three categories was .75 or above. For the final analysis, a Pearson correlation for each dimension was conducted between all possible pairs of raters' responses using all the available pairs. Correlation evaluations for agreement not only provide a measure of typical human evaluation, they also serve to evaluate the

computational model in comparison to such experts. In addition, because the output of the Entailer is a continuous value, correlations are a practical and effective evaluation of the efficacy of the system.

Table 4. Categorization of Responses for the Target Sentence *John drove to the store to buy supplies.*

Category	Student Statement	Relationship to Source Sentence
Entailment	John went to the store.	Explicit, logical implication
Elaboration	He could have borrowed stuff.	Non-contradictory reaction
Paraphrase	He took his car to the store to get things that he wanted.	Reasonable restatement

As Hatch and Lazarton (1991) point out, the more raters used for assessing inter-rater reliability, the greater the confidence of the reliability. Hatch and Lazaraton offer the following formula to convert multiple raters' correlations into a single effective gold inter-rater value:

$$Rtt = \frac{^{nr}AB}{1+(n-1)^{r}AB}$$

In this formula, Rtt is the reliability of all the judges, n corresponds to the number of raters, and ^{r}AB is the average correlation across the raters. Based on this formula, the effective inter-rater reliability for the Pearson correlations were as follows: paraphrase (r = .909), entailment (r = .846), elaboration (r = .819). The *individual* correlations and their averages are given in Table 5.

For this experiment, we removed from the corpus two types of pairings that did not suit this analysis. First, we removed any pair where the response was only garbage (e.g., where the user had randomly hit the keyboard or where responses consisted of no more than one word; such responses would be filtered out of the iSTART system under normal operation). Second, we removed pairs where the target sentence was a question. For these pairs, users had tried to answer the questions rather than self-explain them. Following these removals, our final corpus consisted of 631 pairs. This corpus was further divided into two groups, one for training (419 cases) and one for testing (212 cases).

Table 5: Inter-rater correlations and average for three dimensions of textual similarity

Raters	Paraphrase	Entailment	Elaboration
1-2	0.720	0.595	0.630
1-3	0.793	0.685	0.609
2-3	0.771	0.688	0.604
Average	0.761	0.656	0.614

Predictions. Following previous Entailer comparison studies (e.g., McCarthy et al., 2007), we predicted that Entailer would outperform other textual comparison measures such as LSA. Specifically, for *human paraphrase* evaluations, we predicted Ent-avg to produce the greatest accuracy of evaluation. For *human entailment* evaluations, we predicted that Ent-for would produce the highest accuracy of evaluations. And for *human elaboration* evaluations, we again predicted Ent-for to be the most accurate index. However, if we consider the target

sentence in any pairing to be the *perfect form*, then any paraphrase of that sentence is likely to be longer. The extra length stemming from the students need to express the idea in their own words. Similarly, students elaborating on a sentence may also tend to write responses that are longer than the target sentence. With this in mind, we also hypothesized that the length of the student responses may lead to Ent-rev being the more predictive index of this text comparison for elaborations and paraphrases.

Correlations. The correlation results (based on the *training* data) largely confirmed our predictions (see Table 5). The Entailer's indices produced the highest correlations with human evaluations (paraphrase: r=.818, p<.001; entailment: r=.741, p<.001: elaboration: r=-.673, p <.001). Comparing the Content-overlap and LSA indices, the former was significantly more accurate (paraphrase: *z-diff* = 1.984, p=.047; entailment: *z-diff* = 2.000, p=.045; elaboration: *z-diff*=1.827, p=.068). There was no significant difference between evaluations of LSA, Content-overlap, and MED.

Of the three Entailer indices, the highest correlating index for paraphrase was Ent-rev (r=.818, p<.001). There was no significant difference between this value and that of Ent-avg (r= .769, p<.001), which was our predicted index. We speculate that the apparently higher Ent-rev value results from student responses for paraphrase being longer than their corresponding target sentence.

Given that a target sentence could be considered the *ideal form* of the sentence, a student trying to paraphrase that sentence would probably have to use more words, which indeed they appear to have done when length is compared (F (1, 1334) = 28.686, p<.001).

Regression. Our analysis of the *three* hand-coded text relatedness evaluations (paraphrases, entailment, elaboration) consisted of a series of *forced entry* linear regressions,

selected as a conservative form of multivariate analysis. Regression was selected as the method of analysis because the dependent variables are continuous. One advantage of regression analysis is that derived values generated from B-weights offer a continuous evaluation of each assessment (in this case, 1-6). Ultimately, parameters using this scale can be used to assess optimal ranges that most accurately assess the kind of student input (i.e., an *entailed*, *paraphrased*, or *elaborative* response). The hand coded evaluations of *entailment*, *elaboration*, and *paraphrase* were the dependent variables and the computational evaluation index with the highest correlation to the training set data were used as independent variables. The results below are based on the *test* set data using the coefficients derived from the regressions on the *training* set data.

Table 6: Correlations between comparison type and text evaluation measure (n = 419)

Paraphrase	Ent-Rev	Ent-Avg	Content	MED	LSA	Ent-For
	0.818	0.769	0.659	0.634	0.574	0.566
Entailment	Ent-For	Ent-Avg	MED	Ent-Rev	Content	LSA
	0.741	0.724	0.578	0.577	0.57	0.469
Elaboration	Ent-For	Ent-Avg	Content	MED	LSA	Ent-Rev
	-0.673	-0.576	-0.515	-0.443	-0.416	-0.38

Note: All correlations are significant at $p. < .001$.

Paraphrase. Using Ent-rev as the independent variable, a significant model emerged, $F(1, 417) = 844.151$, $p < .001$. The model explained 66.9% of the variance (Adjusted $R^2 = .669$).

Ent-rev was a significant predictor (t = 29.054, p < .001). The derived B-weights were then used to calculate the accuracy of the model against the held-back *test set* data (n=212). The correlation between the derived evaluation and the hand-coded paraphrase values was high (r=.840, p<.001); indeed, the correlation was significantly higher than that produced by the mean of the human coders (*z-diff* = 2.874, p=.004), suggesting that the model is at least as accurate as the expert raters. When the other indices were added to the model there was no significant increase in accuracy. Replacing Ent-rev with Ent-avg resulted in significantly lower accuracy (r =.755, p<.001; *z-diff* = 2.420, p = .016).

Entailment. Using Ent-for as the independent variable, a significant model emerged, F (1, 417) = 507.936, p< .001. The model explained 54.8% of the variance (Adjusted R^2 = .548). *Ent-for* was a significant predictor (t = 22.537, p < .001). The derived B-weights were then used to calculate the accuracy of the model against the held-back *test set* data (n=212). The correlation between the derived evaluation and the hand-coded entailment values was high (r=.708, p<.001); the correlation was not significantly different from that produced by the human coders, suggesting that the model is at least as accurate as the three expert raters. When the other indices were added to the model there was no significant increase in accuracy.

Elaboration. Using Ent-for as the independent variable, a significant model emerged, F (1, 417) = 345.715, p< .001. The model explained 45.2% of the variance (Adjusted R^2 = .452). *Ent-for* was a significant predictor (t = -18.593, p < .001). The derived B-weights were then used to calculate the accuracy of the model against the held-back *test set* data (n=212). The correlation between the derived evaluation and the hand-coded entailment values was again high (r=.676, p<.001); the correlation was not significantly different from that produced by the human coders,

suggesting that the model is at least as accurate as the three expert raters. When the other indices were added to the model there was no significant increase in accuracy.

6. Discussion

We discussed in this paper the close relationship between the three semantic relations of elaboration, entailment, and paraphrase that can hold between two text fragments. In particular, we focused our discussion on fragments the size of a sentence. We have also presented the basic algorithm for detecting the relationship of entailment called the Entailer (Ent-For) and showed how it could be extened to handle the relations of elaboration (Ent-Rev) and paraphrase (Ent-Avg). A fully implemented system to automatically detect the three semantic relationships was implemented and experimented with. We reported experimental results on two data sets, the Microsoft Research Paraphrase (MSRP) corpus and the iSTART corpus. The two data sets are quite different offering a diverse testing ground for the proposed methods. The MSRP contains professionally written texts from various news sources while the iSTART corpus contains student-typed paraphrases which are with typos and less grammatical.

The results obtained for the original MSRP corpus and for the balanced data set are encouraging. A close analysis of the results obtained on the original MSRP corpus reveals that while the Average Entailer method offers better accuracy and recall it is less powerful than both Forward and Reverse methods in terms of precision. Because recall is better for the Average system than the Forward and Reverse systems, the Average system can detect paraphrases very well. However, the precision of the Average method is slightly lower than Forward and Reverse, indicating that the Average Entailer method results in more false positives than the other two methods. For the balanced data set (50-50 split of TRUE-FALSE instances), the accuracy of the

Average system is still best but, interestingly, the precision and recall pattern changes. That is, the Average system has better precision but slightly worse recall than the Forward and Reverse systems. In this case, the Average system can less successfully detect paraphrases but when it does so, it does it with high precision. This apparent discrepancy between the behavior of the approaches on different data sets could be the result of the different distributions of TRUE and FALSE positive cases in the two data sets. In the original MS paraphrase corpus, the number of positive instances greatly outnumbers the negative instances.

For the iSTART data set, we also compared various indices derived from the Entailer to a variety of other text relatedness metrics (e.g., LSA). Our corpus was formed from 631 iSTART target-sentence/self-explanation pairs. The self-explanations were hand coded across three categories of text relatedness: *paraphrase, entailment,* and *elaboration*. A series of regression analyses suggested that the *Entailer* was the best measure for approximating these hand coded values. The *Ent-rev* index of the *Entailer* explained approximately 67% of the variance for paraphrase; the *Ent-for* index explained approximately 55% of the variance for entailment, and 45% of the variance for elaboration. For each model, the derived evaluations either met or surpassed human inter-rater correlations, meaning that the algorithms can produce assessments of text at least equal to that of three experts.

The accuracy of our proposed methods is highly encouraging. Future work can now move towards implementing algorithms that use these Entailer-based methods to provide feedback to and assess that feedback when applied to users of the iSTART system. As each model produces a value between approximately 1 and 6, we envision that dividing these values into *low* (e.g. <2.67), *moderate* (e.g. >2.67 < 4.33), and *high* (e.g. >4.33) partitions will allow us to provide users with accurate feedback on their input. For example, a moderate paraphrase evaluation,

coupled with a high elaboration evaluation might call for a feedback response such as "*Your paraphrase is fairly good. However, you appear to have included a lot of information that is not really relevant. See if you can write your paraphrase again with more information from the target sentence, and reduce the information that is not in the target sentence.*" While only the Entailer indices contributed to the final assessment models in the ITS experiments, all other measures (e.g., LSA) correlated highly with the hand coded evaluations. This finding is important because these other measures are still envisioned to contribute to a final feedback algorithm. Specifically, a high content-overlap evaluation coupled with a high paraphrase evaluation could indicate that a paraphrase may have been successful only because many of the words from the target sentence were reproduced in the response.

Previous research has shown that the Entailer delivers high performance analyses when compared to similar systems in the industry approved testing ground of *Recognizing Textual Entailment* tasks (HIDDEN REFERENCE, YEAR). However, the natural language input from the ITS corpus used in this study (with its spelling, grammar, asymmetrical, and syntax issues) provided a far sterner testing ground. The results of this study suggest that in this environment too, the performance of the Entailer has been significantly better than comparable approaches. This finding is compelling because highly accurate assessment metrics are necessary to better assess input and supply the most optimal feedback to students. This study offers promising developments in this endeavor.

7. Conclusions

The article discussed the three semantic relations of entailment, elaboration, and paraphrase, and presented an algorithm that could be used to recognize any of the three relations. We fully

implemented the basic Entailer algorithm for handling entailment and its extensions for elaboration and paraphrase. We used the implemented systems to experiment with the proposed algorithms on two data sets that are quite different in their nature. We found that the proposed approach offers competitive results with other approaches on standardized data sets and that for the evaluation of student-generated paraphrases in intelligent tutoring system our method offers results that rival human judgments.

Acknowledgments

HIDDEN PARAGRAPH.

References

Barzilay, R., & McKeown, K. (2001). Extracting paraphrases from a parallel corpus. In *39th Annual Meeting of the Association for Computational Linguistics*, 50–57.

Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., & Buchanan, B.G. (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34, 301–310.

Dagan, I., and Glickman, O. (2004). Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of Learning Methods for Text Understanding and Mining*.

Dagan, I., Glickman, O., & Magnini, B. (2004-2005). Recognizing textual entailment. In *http://www.pascalnetwork.org/Challenges/RTE*.

Dagan, I., Glickman, O., & Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*.

Dolan, W. B., Quirk, C., & Brockett, C. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING 2004*.

Graesser, A. C., Olney, A., Haynes, B., & Chipman, P. (2005). *Cognitive Systems: Human Cognitive Models in Systems Design*. Erlbaum, Mahwah, NJ. chapter AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue.

Ibrahim, A., Katz, B., & Lin, J. (2003). Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing (ACL 2003)*.

Iordanskaja, L., Kittredge, R., & Polgere, A.(1991). *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic. Chapter Lexical selection and paraphrase in a meaning-text generation model.

Kouylekov, M., & Magnini, B. (2005). Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*.

Kozareva, Z., & Montoyo, A. (2006). *Lecture Notes in Artificial Intelligence: Proceedings of the 5th International Conference on Natural Language Processing (Fin-TAL 2006)*. chapter Paraphrase Identification on the basis of Supervised Machine Learning Techniques.

Landauer, T., McNamara, D.S., Dennis, S., & Kintsch, W. (2007). Latent Semantic Analysis: A road to meaning, 2007, Mahwah, NJ:Erlbaum.

Leacock, C., & Chodorow, M. (1998). Combining local context and wordnet sense similarity for word sense identification. In *WordNet: An Electronic Lexical Database*. MIT Press.

Lin, D., & Pantel, P. (2001). Dirt - discovery of inference rules from text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, 323–328.

McCarthy, P., Rus, V., Crossley, S., Bigham, S., Graesser,A., & McNamara, D. (2007). Assessing entailer with a corpus of natural language. In *Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS)*. Menlo Park, CA: AAAI Press.

McNamara, D. S., Boonthum, C., Levinstein, I. B., & Millis, K. (2007). *Handbook of Latent Semantic Analysis*. Mahwah, NJ: Erlbaum. chapter Evaluating selfexplanations in iSTART: comparing word-based and LSA algorithms, 227–241.

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence (AAAI 2006)*.

Miller, G. (1995). WordNet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Monz, C., and de Rijke, M. (2001). *Light-Weight Entailment Checking for Computational Semantics*. 59–72.

Pazienza, M., Pennacchiotti, M., and Zanzotto, F. (2005). Textual entailment as syntactic graph distance: A rule based and svm based approach. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*.

Qiu, L., Kan, M., and Chua, T. (2006). Paraphrase recognition via dissimilarity significance classification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 18–26. Association of Computational Linguistics.