iSTART: Paraphrase Recognition

Chutima Boonthum
Computer Science Department
Old Dominion University, Norfolk, VA-23508 USA
cboont@cs.odu.edu

Abstract

Paraphrase recognition is used in a number of applications such as tutoring systems, question answering systems, and information retrieval systems. The context of our research is the iSTART reading strategy trainer for science texts, which needs to understand and recognize the trainee's input and respond appropriately. This paper describes the motivation for paraphrase recognition and develops a definition of the strategy as well as a recognition model for paraphrasing. Lastly, we discuss our preliminary implementation and research plan.

1 Introduction

A web-based automated reading strategy trainer called iSTART (Interactive Strategy Trainer for Active Reading and Thinking) adaptively assigns individual students to appropriate reading training programs. It follows the SERT (Self-Explanation Reading Training) methodology developed by McNamara (in press) as a way to improve high school students' reading ability by teaching them to use active reading strategies in self-explaining difficult texts. Details of the strategies can be found in McNamara (in press) and of iSTART in Levinstein et al. (2003)

During iSTART's practice module, the student self-explains a sentence. Then the trainer analyzes the student's explanation and responds. The current system uses simple word- matching algorithms to evaluate the student's input that do not yield results that are sufficiently reliable or accurate. We therefore propose a new system for handling the student's explanation more effectively. Two major tasks of this semantically-based system are to (1) construct an internal representation of sentences and explanations and (2) recognize the reading strategies the student uses beginning with paraphrasing.

Construct an Internal Representation: We transform the natural language explanation into a representation suitable for later analysis. The *Sentence Parser* gives us a syntactically and morphologically tagged representation. We transform the output of the Link Grammar parser (CMU, 2000) that generates syntactical and morphological information into an appropriate knowledge representation using the *Representation Generator*.

Recognize Paraphrasing: In what follows, we list the paraphrase patterns that we plan to cover and define a recognition model for each pattern. This involves two steps: (1) recognizing paraphrasing patterns, and (2) reporting the result. The *Paraphrase Recognizer* compares two internal representation (one is of a given sentence and another is of the student's explanation) and finds paraphrase matches ("concept-relation-concept" triplet matches) according to a paraphrasing pattern. The *Reporter* provides the final summary of the total paraphrase matches, noting unmatched information in either the sentence or the explanation. Based on the similarity measure, the report will include whether the student has fully or partially paraphrased a given sentence and whether it contains any additional information.

2 Paraphrase

When two expressions describe the same situation, each is considered to be a paraphrase of the other. There is no precise paraphrase definition in general; instead there are frequently-accepted paraphrasing patterns to which various authorities refer. Academic writing centers (ASU Writing Center, 2000; BAC Writing Center; USCA Writing Room; and Hawes, 2003) provide a number of characterizations, such as using syno-

nyms, changing part-of-speech, reordering ideas, breaking a sentence into smaller ones, using definitions, and using examples. McNamara (in press), on the other hand, does not consider using definitions or examples to be part of paraphrasing, but rather considers them elaboration. Stede (1996) considers different aspects or intentions to be paraphrases if they mention the same content or situation.

Instead of attempting to find a single paraphrase definition, we will start with six *commonly mentioned* paraphrasing patterns:

1. **Synonym**: substitute a word with its synonym, e.g. help, assist, aid;
2. **Voice**: change the voice of sentence from active to passive or vice versa;
3. **Word-Form/Part-of-speech**: change a word into a different form, e.g. change a noun to a verb, adverb, or adjective;
4. **Break down Sentence**: break a long sentence down into small sentences;
5. **Definition/Meaning**: substitute a word with its definition or meaning;
6. **Sentence Structure**: use different sentence structures to express the same thing.

If the explanation has any additional information or misses some information that appeared in the original sentence, we should be able to detect this as well for use in discovering additional strategies employed.

3 Recognition Model

To recognize paraphrasing, we convert natural language sentences into Conceptual Graphs (CG, Sowa, 1983; 1992) and then compare two CGs for matching according to paraphrasing patterns. The matching process is to find as many "concept-relation-concept triplet" matches as possible. A triplet match means that a triplet from the student's input matches with a triplet from the given sentence. In particular, the left-concept, right-concept, and relation of both sub-graphs have to be exactly the same, or the same under a transformation based on a relationship of synonymy (or other relation defined in WordNet), or the same because of idiomatic usage. It is also possible that several triplets of one sentence together match a single triplet of the other. At the end of this pattern matching, a summary result is provided: total paraphrasing matches, unparaphrased information and additional information (not appearing in the given sentence).

3.1 Conceptual Graph Generation

A natural language sentence is converted into a conceptual graph using the Link Grammar parser. This process mainly requires mapping one or more Link connector types into a relation of the conceptual graph.

A parse from the Link Grammar consists of triplets: starting word, an ending word, and a connector type between these two words. For example, [1 2 (Sp)] means word-1 connects to word-2 with a subject connector or that word-1 is the subject of word-2. The sentence "A walnut is eaten by a monkey" is parsed as follows:

```
[(0=LEFT-WALL)(1=a)(2=walnut.n)(3=is.v)
(4=eaten.v)(5=by)(6=a)(7=monkey.n)(8=.)]
[[0 8 (Xp)][0 2 (Wd)][1 2 (Dsu)][2 3 (Ss)]
[3 4 (Pv)][4 5 (MVp)][5 7 (Js)][6 7 (Ds)]]
```

We then convert each Link triplet into a corresponding CG triplet. Two words in the Link triplet can be converted into two concepts of the CG. To decide whether to put a word on the left or the right side of the CG triplet, we define a mapping rule for each Link connector type. For example, a Link triplet [1 2 (S*)] will be mapped to the 'Agent' relation, with word-2 as the left-concept and word-1 as the right-concept: [Word-2] → (Agent) → [Word-1]. Sometimes it is necessary to consider several Link triplets in generating a single CG triplet. A CG of previous example is shown below:

```
0 [0 8 (Xp)]  -> #S#  -> - N/A -
1 [0 2 (Wd)]  -> #S#  -> - N/A -
2 [1 2 (Dsu)] -> #S#  ->
    [walnut.n]->(Article)->[a]
3 [2 3 (Ss)] -> #M# S + Pv (4) # ->
    [eaten.v]->(Patient)->[walnut.n]
4 [3 4 (Pv)] -> #M# Pv +MV(5)+O(6)# ->
    [eaten.v] -> (Agent) -> [monkey.n]
5 [4 5 (MVp)] -> #S#  eaten.v by
6 [5 7 (Js)]  -> #S#  monkey.n by
7 [6 7 (Ds)]  -> #S#  ->
    [monkey.n] -> (Article) -> [a]
```

Each line (numbered 0-7) shows a Link triplet and its corresponding CG triplet. These will be used in the recognition process. The '#S#' and '#M' indicate single and multiple mapping rules.

3.2 Paraphrase Recognition

We illustrate our approach to paraphrase pattern recognition on single sentences: using synonyms (single or compound-word synonyms and idiomatic expressions), changing the voice, using a different word form, breaking a long sentence into smaller sentences, substituting a definition for a word, and changing the sentence structure.

Preliminaries: Before we start the recognition process, we need to assume that we have all the information about the text: each sentence has various content words (excluding such 'stop words' as *a*, *an*, *the*, etc.); each content word has a definition together with a list of synonyms, antonyms, and other relations provided by WordNet (Fellbaum, 1998). To prepare a given text and a sentence, we plan to have an automated process that generates necessary information as well as manual intervention to verify and rectify the automated result, if necessary.

Single-Word Synonyms: First we discover that both CGs have the same pattern and then we check whether words in the same position are synonyms. Example:

"Jenny helps Kay"
[Help] → (Agent) → [Person: Jenny]
+⟶ (Patient) → [Person: Kay]
vs.
"Jenny assists Kay"
[Assist] → (Agent) → [Person: Jenny]
+⟶ (Patient) → [Person: Kay]

Compound-Word Synonyms: In this case, we need to be able to match a word and its compound-word synonym. For example, 'install' has 'set up' and 'put in' as its compound-word synonyms. The compound words are declared by the parser program. During the preliminary processing CGs are pre-generated.

[Install] → (Object) → [Thing]
≡ [Set-Up] → (Object) → [Thing]
≡ [Put-In] → (Object) → [Thing]

Then, this case will be treated like the single-word synonym.

"Jenny installs a computer"
[Install] → (Agent) → [Person: Jenny]
+⟶ (Object) → [Computer]
vs.
"Jenny sets up a computer"
[Set-Up] → (Agent) → [Person: Jenny]
+⟶ (Object) → [Computer]

Idiomatic Clause/Phrase: For each idiom, a CG will be generated and used in the comparison process. For example, the phrase 'give someone a hand' means 'help'. The preliminary process will generate the following conceptual graph:

[Help] → (Patient) → [Person: x]
≡ [Give] → (Patient) → [Person: x]
+⟶ (Object) → [Hand]

which gives us

"Jenny gives Kay a hand"
[Give] → (Agent) → [Person: Jenny]
+⟶ (Patient) → [Person: Kay]
+⟶ (Object) → [Hand]

In this example, one might say that a 'hand' might be an actual (physical) hand rather than a synonym phrase for 'help'. To reduce this particular ambiguity, the analysis of the context may be necessary.

Voice: Even if the voice of a sentence is changed, it will have the same CG. For example, both "Jenny helps Kay" and "Kay is helped by Jenny" have the same graphs as follows:

[Help] → (Agent) → [Person: Jenny]
+⟶ (Patient) → [Person: Kay]

At this time we are assuming that if two CGs are exactly the same, it means paraphrasing by changing voice pattern. However, we plan to introduce a modified conceptual graph that retains the original sentence structure so that we can verify that it was paraphrasing by change of voice and not simple copying.

Part-of-speech: A paraphrase can be generated by changing the part-of-speech of some keywords. In the following example, the student uses "a historical life story" instead of "life history", and 'similarity' instead of 'similar'.

Original sentence: "All thunderstorms have a similar life history."

Student's Explanation: "All thunderstorms have similarity in their historical life story."

To find this paraphrasing pattern, we look for the same word, or a word that has the same base-form. In this example, the sentences share the same base-form for 'similar' and 'similarity' as well as for 'history' and 'historical'.

Breaking long sentence: A sentence can be explained by small sentences coupled up together in such a way that each covers a part of the original sentence. We integrate CGs of all sentences in the student's input together before comparing it with the original sentence.

Original sentence: "All thunderstorms have a similar life history."
[Thunderstorm: ∀] –
(Feature) → [History] –
(Attribute) → [Life]
(Attribute) → [Similar]

Student's Explanation: "Thunderstorms have life history. It is similar among all thunderstorms"
[Thunderstorm] –
(Feature) → [History] –
(Attribute) → [Life]
[It] (pronoun)–
(Attribute) → [Similar]
(Mod) → [Thunderstorm: ∀] (among)

We will provisionally assume that the student uses only the words that appear in the sentence in this breaking down process. One solution is to combine graphs from all sentences together. This can be done by merging graphs of the same concept. This process involves *pronoun resolution*. In this example, '*it*' could refer to '*life*' or '*history*'. Our plan is to exercise all possible pronoun references and select one that gives the best paraphrasing recognition result.

Definition/Meaning: A CG is pre-generated for a definition of each word and its associations (synonyms, idiomatic expressions, etc.). To find a paraphrasing pattern of using the definition, for example, a '*history*' means "*the continuum of events occurring in succession leading from the past to the present and even into the future*", we build a CG for this as shown below:

[Continuum] –
(Attribute) → [Event: ∃]
[Occur] –
(Patient) → [Event: ∃]
(Mod) → [Succession] (in)
[Lead] –
(Initiator) → [Succession]
(Source) → [Time: Past] (from)
(Path) → [Time: Present] (to)
(Path) → [Time: Future] (into)

We refine this CG by incorporating CGs of the definition into a single integrated CG, if possible.

(Patient) → [Event: ∃]
(Mod) → [Succession] (in)
(Source) → [Time: Past] (from)
(Path) → [Time: Present] (to)
(Path) → [Time: Future] (into)

From WordNet 2.0, the synonyms of '*past*', '*present*', and '*future*' found to be "*begin*, *start*, *beginning process*", "*middle*, *go though*, *middle process*", and "*end*, *last*, *ending process*", respectively. The following example shows how they can be used in recognizing paraphrases.

Original sentence: "All thunderstorms have a similar life history."
[Thunderstorm: ∀] –
(Feature) → [History] –
(Attribute) → [Life]
(Attribute) → [Similar]

Student's Explanation: "Thunderstorms go through similar cycles. They will begin the same, go through the same things, and end the same way."
[Go] –
(Agent) → [Thunderstorm: #]
(Path) → [Cycle] → (Attribute) → [Similar]
[Begin] –
(Agent) → [Thunderstorm: #]
(Attribute) → [Same]
[Go-Through] –
(Agent) → [Thunderstorm: #]
(Path) → [Thing: ∃] → (Attribute) → [Same]
[End] –
(Agent) → [Thunderstorm: #]
(Path) → [Way: ∃] → (Attribute) → [Same]

From this CG, we found the use of 'begin', 'go-through', and 'end', which are parts of the CG of history's definition. These together with the correspondence of words in the sentences show that the student has used paraphrasing by using a definition of 'history' in the self-explanation.

Sentence Structure: The same thing can be said in a number of different ways. For example, to say "There is someone happy", we can say "Someone is happy", "A person is happy", or "There is a person who is happy", etc. As can be easily seen, all sentences have a similar CG triplet of "[Person: ∃] → (Char) → [Happy]" in their CGs. But, we cannot simply say that they are paraphrases of each other; therefore, need to study more on possible solutions.

3.3 Similarity Measure

The similarity between the student's input and the given sentence can be categorized into one of these four cases:

1. Complete paraphrase without extra info.
2. Complete paraphrase with extra info.
3. Partial paraphrase without extra info.
4. Partial paraphrase with extra info.

To distinguish between '*complete*' and '*partial*' paraphrasing, we will use the triplet matching result. What counts as complete depends on the

context in which the paraphrasing occurs. If we consider the paraphrasing as a *writing technique*, the 'complete' paraphrasing would mean that all triplets of the given sentence are matched to those in the student's input. Similarly, if any triplets in the given sentence do not have a match, it means that the student is 'partially' paraphrasing at best. On the other hand, if we consider the paraphrasing as a *reading behavior or strategy*, the 'complete' paraphrasing may not need all triplets of the given sentence to be matched. Hence, recognizing which part of the student's input is a paraphrase of which part of the given sentence is significant. How can we tell that this explanation is an adequate paraphrase? Can we use information provided in the given sentence as a measurement? If so, how can we use it? These questions still need to be answered.

4 Related Work

A number of people have worked on paraphrasing such as the multilingual-translation recognition by Smith (2003), the multilingual sentence generation by Stede (1996), universal model paraphrasing using transformation by Murata and Isahara (2001), DIRT – using inference rules in question answering and information retrieval by Lin and Pantel (2001). Due to the space limitation we will mention only a few related works.

ExtrAns (Extracting answers from technical texts) by (Molla et al, 2003) and (Rinaldi et al, 2003) uses *minimal logical forms* (MLF) to represent both texts and questions. They identify *terminological* paraphrases by using a term-based hierarchy with their synonyms and variations; and *syntactic* paraphrases by constructing a common representation for different types of syntactic variation via meaning postulates. Absent a paraphrase, they loosen the criteria by using hyponyms, finding highest overlap of predicates, and simple keyword matching.

Barzilay & Lee (2003) also identify paraphrases in their paraphrased sentence generation system. They first find different paraphrasing rules by clustering sentences in comparable corpora using n-gram word-overlap. Then for each cluster, they use *multi-sequence alignment* to find intra-cluster paraphrasing rules: either morpho-syntactic or lexical patterns. To identify inter-cluster paraphrasing, they compare the slot values without considering word ordering.

In our system sentences are represented by conceptual graphs. Paraphrases are recognized through idiomatic expressions, definition, and sentence break up. Morpho-syntatic variations are also used but in more general way than the term hierarchy-based approach of ExtrAns.

5 Preliminary Implementation

We have implemented two components to recognize paraphrasing with the CG for a single simple sentence: *Automated Conceptual Graph Generator* and *Automated Paraphrasing Recognizer.*

Automated Conceptual Graph Generator: is a C++ program that calls the Link Grammar API to get the parse result for the input sentence, and generates a CG. We can generate a CG for a simple sentence using the first linkage result. Future versions will deal with complex sentence structure as well as multiple linkages, so that we can cover most paraphrases.

Automated Paraphrasing Recognizer: The input to the Recognizer is a pair of CGs: one from the original sentence and another from the student's explanation. Our goal is to recognize whether any paraphrasing was used and, if so, what was the paraphrasing pattern. Our first implementation is able to recognize paraphrasing on a single sentence for exact match, direct synonym match, first level antonyms match, hyponyms and hypernyms match. We plan to cover more relationships available in WordNet as well as definitions, idioms, and logically equivalent expressions. Currently, voice difference is treated as an exact match because both active voices have the same CGs and we have not yet modified the conceptual graph as indicated above.

6 Discussion and Remaining Work

Our preliminary implementation shows us that paraphrase recognition is feasible and allows us to recognize different types of paraphrases. We continue to work on this and improve our recognizer so that it can handle more word relations and more types of paraphrases. During the testing, we will use data gathered during our previous iSTART trainer experiments. These are the actual explanations entered by students who were given the task of explaining sentences. Fortu-

nately, quite a bit of these data have been evaluated by human experts for quality of explanation. Therefore, we can *validate* our paraphrasing recognition result against the human evaluation.

Besides implementing the recognizer to cover all paraphrasing patterns addressed above, there are many issues that need to be solved and implemented during this course of research.

The *Representation* for a simple sentence is the Conceptual Graph, which is not powerful enough to represent complex, compound sentences, multiple sentences, paragraphs, or entire texts. We will use *Rhetorical Structure Theory* (RST) to represent the relations among the CGs of these components of these more complex structures. This will also involve *Pronoun Resolution* as well as *Discourse Chunking*. Once a representation has been selected, we will implement an automated generator for such representation.

The *Recognizer* and *Paraphrase Reporter* have to be completed. The *similarity measures* for writing technique and reading behavior must still be defined.

Once all processes have been implemented, we need to verify that they are correct and validate the results. Finally, we can integrate this recognition process into the iSTART trainer in order to improve the existing evaluation system.

Acknowledgements

This dissertation work is under the supervision of Dr. Shunichi Toida and Dr. Irwin Levinstein. iSTART is supported by National Science Foundation grant REC-0089271.

References

ASU Writing Center. 2000. Paraphrasing: Restating Ideas in Your Own Words. Arizona State University, Tempe: AZ.

BAC Writing Center. Paraphrasing. Boston Architectural Center. Boston: MA.

Carnegie Mellon University. 2000. Link Grammar.

R. Barzilay and L. Lee. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *HLT-NAACL*, Edmonton: Canada, pp. 16-23.

C. Boonthum, S. Toida, and I. Levinstein. 2003. Paraphrasing Recognition through Conceptual Graphs. Computer Science Department, Old Dominion University, Norfolk: VA. (TR# is not available)

C. Boonthum. 2004. iSTART: Paraphrasing Recognition. *Ph.D. Proposal*: Computer Science Department, Old Dominion University, VA.

C. Fellbaum. 1998. *WordNet: an electronic lexical database*. The MIT Press: MA.

K. Hawes. 2003. Mastering Academic Writing: Write a Paraphrase Sentence. University of Memphis, Memphis: TN.

I. Levinstein, D. McNamara, C. Boonthum, S. Pillarisetti, and K. Yadavalli. 2003. Web-Based Intervention for Higher-Order Reading Skills. In *ED-MEDIA*, Honolulu: HI, pp. 835-841.

D. Lin and P. Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering* 7(4):343-360.

W. Mann and S. Thompson, 1987. Rhetorical Structure Theory: A Theory of Text Organization. *The Structure of Discourse*, Ablex.

D. McNamara. (in press). SERT: Self-Explanation Reading Training. *Discourse Processes*.

D. Molla, R. Schwitter, F. Rinaldi, J. Dowdall, and M. Hess. 2003. ExtrAns: Extracting Answers from Technical Texts. *IEEE Intelligent System* 18(4): 12-17.

M. Murata and H. Isahara. 2001. Universal Model for Paraphrasing – Using Transformation Based on a Defined Criteria. In *NLPRS: Workshop on Automatic Paraphrasing: Theories and Application*.

F. Rinaldi, J. Dowdall, K. Kaljurand, M. Hess, and D. Molla. 2003. Exploiting Paraphrases in Question Answering System. In *ACL: Workshop in Paraphrasing*, Sapporo: Japan, pp. 25-32.

N. Smith. 2002. From Words to Corpora: Recognizing Translation. In *EMNLP*, Philadelphia: PA.

J. Sowa. 1983. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, MA.

J. Sowa. 1992. Conceptual Graphs as a Universal Knowledge Representation. *Computers Math. Application*, 23(2-5): 75-93.

M. Stede. 1996. *Lexical semantics and knowledge representation in multilingual sentence generation.* Ph.D. thesis: Department of Computer Science, University of Toronto, Canada.

USCA Writing Room. *Paraphrasing*. The University of South Carolina: Aiken. Aiken: SC.