Exploiting Connectivity for Case Construction in Learning by Reading

David Barbella BARBELLA@U.NORTHWESTERN.EDU
Kenneth D. Forbus FORBUS@NORTHWESTERN.EDU
Qualitative Reasoning Group, EECS Department, 2133 Sheridan Rd, Evanston IL 60208USA

Abstract

One challenge faced by cognitive systems is how to organize information that is learned by reading. Analogical reasoning provides a method for immediately using learned knowledge, and analogical generalization potentially provides a means to integrate knowledge across multiple sources. To use analogy requires organizing information into effective cases. This paper argues that using connectivity in semantic interpretations to organize knowledge learned by reading into overlapping cases can support analogical reasoning with learned knowledge. Two connectivity-based methods are described, and their performance is compared with two baselines for the task of comparing and contrasting topics included in material the system has read.

1. Introduction and Motivation

Cognitive systems need to learn from reading in order to acquire knowledge in a scalable way. Solid progress has been made in natural language understanding and knowledge representation that enables complex structured information to be extracted from text (e.g. Barker et al., 2007; Fan et al., 2012). A challenge this raises is how such extracted knowledge can be organized for reasoning and integration into what a cognitive system already knows. Analogical reasoning has been shown to be useful for robust learning by reading, for decoding instructional analogies (Barbella & Forbus, 2011), and for allowing a system to ask itself questions based on analogies with prior knowledge (Forbus et al., 2007).

Case comparison has been used or proposed for use in other reasoning applications as well (Brüninghaus & Ashley, 2001; Peterson, Mahesh, & Goel, 1994; Chaudhri et al., 2014). This suggests that one way of organizing newly read knowledge is to group it into *cases* – groups of facts, treated as a unit – so that it can be used in analogical reasoning and learning. Analogy works best with interconnected relational structure, where conceptually related information is in the same case. This suggests going beyond the natural boundaries provided by language – paragraphs and sentences – and focusing instead on interconnected facts within the conceptual representations produced by semantic interpretation.

This paper describes two connectivity-based segmentation algorithms and evaluates them by comparison with sentence and paragraph level algorithms. We begin by summarizing the systems and representations the system uses. We then introduce four methods for organizing facts into cases. Two methods use only natural text boundaries (sentences and paragraphs), and two are connection-based. We follow with a description of an experiment and its results, and close with related and future work.

2. Background

This section summarizes our learning by reading system, analogical matching, and case construction. We discuss each in turn.

Our learning by reading system is based on the Companion cognitive architecture (Forbus, Klenk, & Hinrichs, 2009), which provides reasoning facilities (including analogical reasoning) that are heavily used during language processing. We use the Cyc representation language, ontology, and knowledge base contents[1], which provide a large vocabulary of concepts (called *collections*), predicates, and several million facts constraining them. We use fixed-width font to denote symbols and expressions from the system, e.g. `(isa solar-panel02 SolarCollector)` says that the entity `solar-panel02` is an instance of the collection (i.e. concept) `SolarCollector`. Cyc's language also provides a notion of logical environment via *microtheories*, local contexts linked via inheritance relationships. Cyc's inclusion of rich type-level representations as well as microtheories make it a natural choice for expressing the kinds of complex information often communicated via language.

Our language processing pipeline uses Allen's (1994) parser for syntactic analysis. Syntactic and lexical ambiguities are explicitly encoded as choice sets. Our semantic interpretation process is based on *Discourse Representation Theory* (DRT; Kamp & Reyle, 1993), which provides an account of scoping, including conditions and counterfactuals, and is central to one of our segmentation techniques. Semantic interpretation builds up *discourse representation structures* (DRSs), each of which is a case containing one or more facts. These facts describe relationships between entities, collections that those entities belong to (i.e. category information), and other DRSs. Each sentence has an associated sentence DRS that contains the facts that represent the semantics of that sentence. Consider "The solar panel cools." The sentence DRS constructed for it contains three facts. The first, `(isa solar-panel02 SolarCollector)`, indicates that there is a solar panel. The symbol `solar-panel02` is a discourse variable created by the system to represent the solar panel being discussed. The `02` after the name is an integer appended to ensure uniqueness, in the event that multiple solar panels are mentioned (the actual ids are more complex, but we simplify here for conciseness). The name `solar-panel02` is derived from the span of text to be human-readable, but means nothing on its own to the reasoning system. The system knows that it is a solar panel because the `isa` relationship indicates that it is `SolarCollector`, which is a collection in the knowledge base (i.e., a concept). `(isa cool05 CoolingEvent)` is similar. The system reifies events and uses role relations to tie the participants in those events to them, as in `(objectOfStateChange cool05 solar-panel02)`, the third fact generated for the sentence representation. These facts were generated by frames encoding links between linguistic and conceptual knowledge in the KB. The first fact was generated by a frame for the multiword string "solar panel," and the second and third were generated by a frame for the word "cool." Where there are multiple possible semantic representations, the system represents these as choice sets over the different possibilities.

[1] Here, ResearchCyc (www.cyc.com)

Sentence DRSs can also have constituent DRSs. For example, the sentence "If the valve closes, the flow of water stops" mentions a closing event and a stopping event, but does not say that either actually happened. If the system produced facts that said that a stopping event occurred and that the flow of water was what stopped, it would reach incorrect conclusions. For that reason, rather than placing that information directly in the sentence DRS, the system creates two new constituent DRSs, one for the antecedent of the statement and one for the consequent. The sentence DRS then contains a fact of the form `(implies DRS-01 DRS-02)`. Constituent DRSs are also used to handle negation and hypotheticals.

The process of semantic interpretation involves constructing DRSs by finding solutions for the syntactic and lexical choices. We use multiple strategies to select among the available choices. To factor out any domain or task specific influences, ambiguities were resolved automatically using a small set of general-purpose heuristics. For example, the reading system preferred interpretations that treat compound noun phrases like "solar panel" as atomic referents when they are available, i.e. `SolarPanel`. An interpretation that interpreted "solar panel" in a more general sense – as a generic panel that is in some way related to the sun – would be less preferred. Another heuristic prefers interpretations that include more facts over interpretations that include fewer. The final fallback heuristic is to choose randomly.

Given that these heuristics do not involve any learned statistics and are task independent, they are surprisingly good. Overall, they selected a correct answer for 86.6% of the 886 lexical choice sets generated by the source texts used in this paper. Errors generated by the heuristics tend to be systematic, which helps prevent analogical processing mismatches. For example, the system consistently selected `Pipe-SmokingDevice` over `Pipe-GenericConduit` as the interpretation of the word "pipe." This was incorrect; the corpus discusses rainwater collection systems and solar heating, but it does not discuss smoking. In the first occurrence of "pipe", the only applicable heuristic was random choice, and smoking device was selected. Subsequent choices were influenced by a heuristic that prefers concepts already used in the interpretation. Hence very similar, albeit partially incorrect, structures were created, facilitating analogical comparison.

After parsing a source text, the system runs a discourse-level interpretation process that handles coreference resolution. The basic strategy used by the coreference system is to resolve pronouns,

Table 1. The simplified version of one paragraph from the corpus.

At the start of each day, the solar heating system is semifull.
At the start of each day, the rainwater collection system is semifull.
The rainwater collection system could have some rainwater in it.
The solar heating system's heat storage is warm because the solar heating system collected heat during the previous day.
In the rainwater collection system, the valve closes.
This prevents the flow of the stored water.
The rain is not falling.
The water in the pipe has leaked out.
The sun has not yet risen on the solar heating system.
The solar collector was exposed to the cold air in the night.
The heat storage contains heat.
Because of this, the heat storage's temperature is greater than the air's temperature.
The control device sensed that the heat storage's temperature was greater than the solar collector's temperature.
Because of this, the control device shut off the pump.
This prevented the cooling process of the heat storage.

definite references ("the dolphin"), and verbs to the most recent valid referent, as determined by common collection membership and a few other factors, such as sharing arguments, in the case of verb coreference. A *discourse* is a multi-sentence section of a source text considered all together. For the work described here, a multi-paragraph chapter from a book and a web encyclopedia article were used as the discourses. After coreference resolution, the system places the contents of the sentence DRSs from the text into a *discourse interpretation* DRS, with coreferent items resolved to the same symbol. This DRS frequently has many constituent DRSs, as every constituent DRS from one of the component sentences becomes a constituent DRS in the discourse interpretation.

For our experiments, the source text was simplified syntactically (Kuehne & Forbus, 2004). This process consists of converting unsupported grammatical structures into supported ones. It does not completely eliminate syntactic ambiguity. For example, the source sentence "As for the solar heater, the sun has not yet risen" was simplified to "The sun has not yet risen on the solar heating system." Table 1 contains a paragraph from the simplified corpus. The segmentation process places no broad-ranging restrictions on the lexicon, although there are occasional coverage gaps, particularly where compound nouns are concerned – for example, the unsupported "solar heater" became "solar heating system" in the sentence above.

Our evaluation involves comparing and contrasting generated cases using analogy. This uses the Structure Mapping Engine (SME; Falkenhainer, Forbus, & Gentner, 1989), a computational implementation of Gentner's (1983) structure mapping theory. SME takes two structured representations, the base and the target, as input. These are the two cases that will be aligned with each other. It produces one or more *mappings*, which consist of three parts. First, mappings include a set of *correspondences* between elements of the base and elements of the target. Second, they include a *score*, which is an estimate of match quality. SME attempts to produce the largest mappings it can that satisfy the constraints of structure-mapping theory. Each mapping also includes *candidate inferences*, hypotheses formed by filling out the target with parts of the base not represented in the target and vice versa. For our evaluation, the score is used to determine which mapping is the best. The correspondences of that mapping can be thought of as things that the cases have in common, and the candidate inferences can be thought of as salient differences between the two cases. Because SME can project inferences in both directions, which case is the base and which is the target is immaterial.

Dynamic case construction (Mostek, Forbus, & Meverden, 2000) is the process of building cases automatically from a body of knowledge. Almost all case-based reasoning systems require cases to be constructed by some external process. These are sometimes hand-curated, but this does not scale well. The ability to build focused cases from larger knowledge sources avoids manual curation, especially when combined with natural language understanding. The methods we describe in this paper – particularly fact-based segmentation - can be thought of as extensions of dynamic case construction for learning by reading. Specifically, the case construction methods all work by starting with a *seed* – an entity or concept about which relevant facts are to be gathered – and proceed by collecting statements in the knowledge base that mention the seed, filtering by those facts, and recursing outward from other entities mentioned in those facts, out to some depth limit. Instead of going to the knowledge base, here we gather facts from the interpretation of a text.

3. Algorithms

To build cases from semantic representations, we developed two connection-based algorithms that make use of properties of the knowledge in the interpretation, and two simpler algorithms that use only naturally-occurring boundaries in text to serve as baselines. All four algorithms start with the same input. The system reads a chapter from a source text and produces the discourse interpretation. Regardless of algorithm, each case is built around a *seed*, which is a single mention of a single entity in the source text. For example, the term "solar heating system" in the sentence "At the start of each day, the solar heating system is semifull" was one seed that was used, and we will use this seed as an example throughout this section. The system generates cases for each possible seed (i.e., each entity that is mentioned) in the source text.

We start with the baselines, as they are simpler. The simplest algorithm is *Local Sentence Interpretation* (LSI), which uses the interpretation of the sentence that mentions the seed (including any constituent DRSs) as the case. This makes intuitive sense as a baseline, because most English-language sentences are generally about a single thing. It is also inexpensive, because no additional computation is required beyond what goes into producing the sentence interpretation in the first place. However, a limitation of this method is that important information about an object or situation is often spread over multiple sentences.

The second baseline algorithm, *Local Paragraph Interpretation* (LPI), is similar, but uses all of the facts derived from all of the sentences in the paragraph of the source text that the seed appears in. The local paragraph interpretation method is much less likely to miss important information about the seed, but it has two potential disadvantages. First, a paragraph can cover multiple topics, which increases the amount of noise in the case. Second, it may be less useful for comparing two seeds from the same paragraph because the cases will be identical, and SME will tend to match most facts with themselves. The match is done with a requirement that the seed entities, which are not identical, align, which helps. For our example seed, the case created by this algorithm is very large and covers a broader range of topics, as it contains the facts built from all 15 sentences in the paragraph.

The last two methods we propose exploit connectivity properties of the conceptual representation of the semantics of the sentence. The first is S*entence-Based Segmentation* (SBS), which works as follows:

1. Find each sentence that the entity is mentioned in.
2. For each of these sentences, find the facts in the discourse interpretation derived from those sentences, including the constituent DRSs. This may include facts that do not mention the entity itself. Add these facts to the case, preserving DRS membership.

Table 2. The facts that were included in a case built using the Sentence-Based Segmentation method. 28 facts were included in total, across 4 DRSs. Representations simplified for space and readability. Note that some facts, such as the ones that include `ObservenceDay` and `Surgery`, are incorrect interpretations. Because the interpretations are generated completely automatically, they contain some word sense errors. The facts shown here were derived from that sentence and from "We examined the elements of a solar heating system" and "At the start of each day, the solar heating system is semifull."

Holds in `Discourse-DRS-01`:
```
(evaluee-Direct compare02 rainwater-collection-system03)
(evaluee-Direct compare02 solar-heating-system01)
(evaluee-Direct examine04 element05)
(fullnessOfContainer solar-heating-system01 PartiallyFull)
(implies-DrsDrs DRS-02 DRS-03)
(isa compare02 Comparing)
(isa day06 ObservanceDay)
(isa examine04 Inspecting)
(isa group-of-element05 Set-Mathematical)
(isa solar-heating-system01 SolarHeatingSystem)
(performedBy compare02 we07)
(performedBy examine04 we08
(possessiveRelation day06 start09)
(startingPoint day06 start09)
(temporallyIntersects start09 (StartFn be10))
(willBe DRS-04)
```

Holds in `DRS-02`:
```
(member element05 group-of-element05)
```

Holds in `DRS-03`:
```
(isa element05 ElementStuffTypeByNumberOfProtons)
(isa solar-heating-system01 SolarHeatingSystem)
(possessiveRelation solar-heating-system01 element05)
```

Holds in `DRS-04`:
```
(conceptuallyRelated day11 Normal-Usual)
(evaluee-Direct examine04 solar-heating-system01)
(isa solar-heating-system01 SolarHeatingSystem)
(performedBy examine04 we08)
(temporallySubsumes day11 operate12)
(isa day11 ObservanceDay)
(isa examine04 Inspecting)
(isa operate12 Surgery)
```

Table 2 contains the 28 facts, drawn from three sentences, that sentence-based segmentation produced using our example seed.

The second connection-based case construction method we propose is *Fact-Based Segmentation* (FBS). This method is inspired by the method described in Mostek et al. (2000), but it has been adapted to use the interpretations produced by the language system. It operates as follows:

1. Choose a maximum depth, i.e. how far the algorithm will travel when recursively gathering facts in the interpretation. A depth of 1 only includes the facts which explicitly mention the

seed, a depth of 2 adds all facts that mention entities mentioned in the facts at depth 1, and so on.

2. Add the seed entity to the case with a depth of 0. (Depth counts up, until the maximum depth is reached.)
3. Identify the collections that the entity belongs to, by examining the `isa` statements that mention it, and add those to the case plus the `isa` statement itself. For each entity in the same paragraph that is a member of that collection, add that entity at depth +1.
4. Identify the other facts that mention the entity. Add them to the case with a depth equal to the entity's depth +1.
5. If the entity is a constituent DRS, add the facts it contains to the case at the same depth. This ensures that if a DRS is mentioned, it is included in the case.
6. For each constituent DRS that contains a fact in the case, add that constituent DRS and the other facts it contains to the case at the same depth.
7. For each entity mentioned in one of the facts added in step 4-6, add that entity to the case at a depth equal to the depth of the fact that mentions it.
8. For each entity added in step 3-7, repeat steps 3-7, stopping when it is not possible to add anything else to the case without exceeding the maximum depth.

Table 3. A subset of the facts in a case constructed via Fact-Based Segmentation. 48 facts were added in total, across 8 DRSs. Representations simplified for space and readability.

```
In Discourse-DRS-01:
(isa solar-heating-system01 SolarHeatingSystem)
(fullnessOfContainer solar-heating-system01 PartiallyFull)
(isa flow15431 FluidFlow-Translation)
(isa prevent15 (PreventingFn flow16))
(primaryObjectMoving flow16 water17)
(not DRS-08)

In DRS-08:
(isa rise13 AscendingEvent)
(isa solar-heating-system01 SolarHeatingSystem)
(objectMoving rise13 sun14)
(on-UnderspecifiedSurface sun14 solar-heating-system01)
```

While this method has several steps, what it does is relatively straightforward. It begins with a seed, like the other methods. Recall our example seed, the term "solar heating system" in the sentence "At the start of each day, the solar heating system is semifull." The algorithm begins by including the discourse variable derived from that seed – `solar-heating-system01` – in the case. When an entity is mentioned in the case, facts that mention that entity are added to the case. This means that the facts `(fullnessOfContainer solar-heating-system01 PartiallyFull)` and `(isa solar-heating-system01 SolarHeatingSystem)` are added to the case, in the contexts they appear in. The first appears in the top-level DRS, `Discourse-DRS-01`, and the second also appears in several constituent DRSs, including `DRS-08`. When a collection is mentioned in the case, entities mentioned in the paragraph that belong to that collection are added to the case, e.g. if there were other instances of `SolarHeatingSystem`, they would be added. When a constituent DRS is mentioned in the

case, facts that mention that constituent DRS and facts inside that constituent DRS are added to the case. Here, when `DRS-08` is added to the case, `(not DRS-08)` is added to the case. When a fact contained in a constituent DRS is in the case, the rest of that constituent DRS is added to the case. Because the case contains a fact in `DRS-08`, this rule adds the other facts in `DRS-08`.

The depth costs results in constituent DRSs (but not top-level sentence DRSs) being added to cases all at the same depth. Adding a complete constituent DRS is crucial for accuracy. For example, consider the sentence "When the temperature of the heat storage equals the temperature of the solar panel, the heat does not flow." Leaving out any part of the DRS constituent structure, e.g. the antecedent information or the negation, would change the meaning of the sentence.

A depth of 3 was chosen based on examinations of pilot data, as reasonable tradeoff between including too much versus too little information. For our example seed, a depth of 3 yielded a case with 48 facts (see Table 3), while a depth of 2 yielded a case of 37 facts and a depth of 4 yielded a case of 60 facts.

We expect that the advantage of this method is that it incorporates connections across multiple sentences. This is potentially useful in cases where a topic is only mentioned once, but is elaborated upon across several sentences. While some sentences that elaborate on a topic may continue to refer to it directly, there may be relevant information in sentences that do not. For example, in "The dolphin often has one calf. The calf is weaned after one year," if the seed being used is the instance of "dolphin" in the first sentence, sentence-based segmentation will not include any information from the second sentence, as it does not mention the dolphin.

4. Evaluation

For evaluation we use a compare and contrast task, as this is one of the simpler kinds of analogical reasoning, and it has interesting potential applications (Brüninghaus & Ashley, 2001; Peterson et al., 1994; Chaudhri et al., 2014). Given two entities in the text, the system compares and contrasts cases generated using those entities as seeds. For differences, we use the notion of *alignable differences* (Gentner & Gunn, 2001), which have been shown to be psychologically salient. These are differences between cases that are conceptually related to their commonalities. In SME, candidate inferences provide alignable differences. Comparisons are evaluated against a hand-generated gold standard of lists of important similarities and differences, generated beforehand. We created two corpora from pre-existing source texts, described next.

The first source text chosen was chapter 16 of *Sun Up to Sun Down* (SUSD; Buckley, 1979), a book about solar energy and solar heating that makes extensive use of analogies. Chapter 16 was chosen because it uses an extended analogy to explain a solar heating system in terms of a rainwater collection system. The simplified version of the chapter consists of 80 sentences in 11 paragraphs (Table 1 shows one paragraph). The interpretation process produced 733 facts across 54 DRSs.

The second source text chosen was a Diffen article that compares and contrasts dolphins and porpoises (Dolphin vs. Porpoise). Diffen is an online, user-editable encyclopedia whose articles compare and contrast similar topics. This particular article was chosen because it differed in both style and subject matter from the other source text. After simplification, the article was 8 paragraphs and 88 sentences long. The interpretation process produced 751 facts across 150 DRSs.

Pairs of seeds were chosen for which the similarities and differences would be illuminating, based on the information available in the texts. For example, after reading the SUSD text, the system was tasked with contrasting the state of a solar heating system at different points during

the day and comparing the solar heating system to an analogous rainwater collection system. All of the tasks related to the Dolphin/Porpoise text involved comparing different aspects of the two creatures, such as their physical anatomy or mating habits. While the system is capable of comparing any pair of objects or events to each other, in most cases arbitrary comparisons are not very interesting. In total, 24 comparisons were made, 11 from the SUSD text and 13 from the Dolphin/Porpoise text. In every comparison, the seeds were required to correspond, i.e. every mapping that SME produced had to put them into alignment.

For each comparison, 2 to 15 *goal facts* were written per comparison, prior to running any of the methods over them. Across the 24 comparisons, 125 goal facts were used in total. 70 of these came from the SUSD text, and 55 from the Dolphin/Porpoise text. Each goal fact represented one similarity or difference in the text. The score for a method, given a pair of cases, was equal to the number of goal facts that it found. Similarities were scored if the similarity was among the correspondences produced by SME. Differences were scored if the difference was among the candidate inferences produced by SME. For example, when cases produced from our example seed were compared to a rainwater collector system seed, all of the methods produced a correspondence between

```
(holdsIn Discourse-DRS-01
(fullnessOfContainer solar-heating-system01 PartiallyFull))
```

and

```
(holdsIn Discourse-DRS-08
(fullnessOfContainer rainwater-collection-system01 PartiallyFull)
```

(These facts were derived from "At the start of the day, the solar heating system is semifull" and "At the start of the day, the rainwater collection system is semifull," respectively.) This indicates that the system could tell that one of the commonalities between the rainwater collection system and the solar heating system, in the scenario being described, is that they're both partially full. The `holdsIn` predicate indicates that the facts are true in the indicated DRS. The goal facts were constructed compositionally to reward more complete answers while still providing some credit for partial answers. For example, rather than using "In the second case, heat flows from the solar collector to the storage tank" as a single example of differences between two cases, it is broken into three statements: One saying that a flow of heat exists, which appears as a fact of the form `(objectMoving flow01 heat15)`, one saying that that flow leaves from the solar collector, which appears as a fact of the form `(fromLocation flow01 solar-collector24)`, and one saying that flow goes to the storage tank, which appears as a fact of the form `(toLocation flow01 storage-tank35)`. Simple features were not included as goal facts to be found. For example, when comparing a solar heating system's operation at different times of the day, the fact that it is a solar heating system in both cases was not among the goal facts. All goal facts were relationships between two entities.

Alternate evaluation metrics were considered. Simply looking for the presence of important facts in cases would not be effective, because facts must show up in both cases in a comparison and must be alignable to be useful. We also compute *generation efficiency*, which is the percentage of correspondences and candidate inferences that were used to produce goal facts. This penalizes including facts that were not useful for this compare and contrast task, although such facts might be useful for other tasks.

Table 4. Experimental Results

Method	LSI	LPI	SBS	FBS
Total Correct	27	81	59	88
Correct (%)	21.6	64.8	47.2	70.4
Generation Efficiency (%)	3.6	1.1	4.7	1.9
Unique Correct	0	8	3	8
Avg. Case Size	8.9	107.2	16.1	66.8
Average CIs	8.4	49.9	19.5	52.0
Average Corrs.	19.6	222.3	29.9	118.6

The results appear in Table 4. The results from the two source texts have been combined, as they were generally comparable across the board. Each column presents results for one of the methods described earlier – Local Sentence Interpretation (LSI), Local Paragraph Interpretation (LPI), Sentence-based segmentation (SBS), and Fact-based Segmentation (FBS). *Total Correct* is the number of the goal facts the method produced. *Correct (%)* is the percent it got correct.

Unique Correct is the number of goal facts where the method was the only one of the four that correctly produced it, a measure of each method's ability to produce interesting conclusions that the others did not. In some instances, including most instances when LPI was the only method to produce a goal fact, it is because the other methods did not include the relevant facts. In other instances, such as the when SBS was the only method to produce the goal fact, one or more of the other methods did include the relevant facts, but failed to produce the proper correspondence or candidate inference.

Avg. Case Size is the average size of the bases and targets produced by the system when constructing cases based on the seeds. As there is no functional difference between bases and targets for this task (candidate inferences are produced in both directions), their sizes are simply averaged together in the table. *Average CIs* and *Average Corrs* are the average number of candidate inferences and correspondences, respectively, produced by the system when comparing cases generated using the method. The number of correspondences produced is fairly high compared to size of the cases because it includes not only fact correspondences, but entity and predicate correspondences. Two complex facts that align with each other can be made up of several different correspondences, as their constituents must also be in alignment, according to structure-mapping theory.

The accuracy of FBS and LPI were very similar. While the two methods found different sets of goal facts, the difference in their overall performance was not statistically significant. The difference between the performance of LSI and the other three methods was statistically significant ($p < 0.005$), as was the difference between SBS and the other three. In total, 100 of the 125 goal facts (80%) were produced by at least one of the methods.

One of the theoretical weaknesses of LPI is that, because it produces pairs of cases that contain all of the same facts when operating on pairs of seeds from the same paragraph, its performance may suffer in those instances. Because the seeds (which are different) are automatically mapped to each other, it can (and did) still produce some useful correspondences and candidate inferences. However, we might suspect that it may be disadvantaged. To test this, we looked at the results on only the comparisons made between entities in the same paragraph. In total, 65 goal

facts came from comparisons of this type. The accuracy on just those goal facts is shown in Table 5. LPI's performance is worse than on the full set, but the difference is not statistically significant, and the difference between LPI and FBS on this limited set is also still not significant.

One of the difficulties with evaluating the system's overall performance is that assigning blame for failures is not easy. For the 20% of the goal facts that no method produced, there are several possible sources of error. First, the goal fact might be one that could be produced by a correctly assembled case, but was not produced by the cases generated by any of the methods used in the experiment. When this is true, further improvements to the case constructor methods could produce better results.

Table 5. Experimental results on cases where the base and target seeds are in the same paragraph.

Method	LSI	LPI	SBS	FBS
Total Correct	20	35	33	41
Correct (%)	30.8	53.8	50.8	63.1

Second, if the formats of the facts are sufficiently different so that SME cannot align them properly, then matches will be suboptimal. This can occur if, for example, what should be similar semantic frames are represented in different ways. Repairing such issues would require rerepresentation (Yan, Forbus, & Gentner, 2003). Table 6 summarizes the sources of error for each of the methods. Each goal fact missed by a method was assigned a source of error, which sometimes varied from method to method. For example, LSI can miss a similarity goal fact by failing to include the facts in the cases to begin with, while FBS can miss the same goal fact by including the corresponding facts, but they don't align when the cases are compared via SME. Some types of errors can occlude other types. For example, if a case construction method fails to include the relevant facts when building the case, it is impossible for those facts to end up being mismatched in the SME mapping.

Table 6. Sources of error.

Method	LSI	LPI	SBS	FBS
Facts Not In Interpretation	7	7	7	7
Facts Not In Case	85	1	49	11
SME Representation Mismatch	2	4	2	4
Wrong Interpretation Choice	2	2	4	3
SME Alignment Mismatch	2	27	6	12

Facts Not In Interpretation means that the facts required to build a particular correspondence or contrast were not present in the global interpretation at all. This error occurred when a goal fact was looking for a fact that could be inferred from the text, but which was not explicitly in the text. For example, causation is not always explicitly spelled out. Even if it could be inferred from context, the system does not currently do this additional reasoning. Because the absence of a key fact from the interpretation makes it impossible for any of the methods to produce a correct response, the same seven goal facts were missed by each of the methods as a result of this error. An example of this occurred when the system was asked to compare the rainwater collection system when it is only lightly raining to the solar heating system early in the morning, when only

a little sunlight falls onto its solar panel. One of the goal facts was "There is a reason that only a little bit of rain/sunlight is collected." While this is something that could be inferred from the text by a human reader, the text does not make an explicit connection between the low amount of incoming rain/sun and the fact that the systems don't collect very much. As a result, there are no causation facts in the discourse interpretation to be aligned. No matter what combination of other facts a segmentation method includes, SME cannot align things that are not there.

Facts Not In Case refers to when key facts were in the global interpretation, but not included by the case construction method. Unsurprisingly, the methods that build larger cases were less likely to produce such errors. This error accounts for the vast majority of the instances where LSI failed to produce a result, as any goal fact that needed information from another sentence would fail.

SME Representation Mismatch refers to when facts representing a similarity were present in the base and target, but were sufficiently different representationally that SME did not match them together. Note that while LPI and FBS missed more goal facts as a result of this error, they are not more likely to produce representation mismatches. All four case construction methods build from the same global interpretation, and all use the same representations as a result. The reason that LPI and FBS had more mismatches is that sometimes LSI and SBS did not include any relevant facts at all. An example of where this occurred was when the system compared the rainwater collection system to the solar heating system while both are operating. The rainwater collection system is collecting rainwater, and the solar heating system is collecting heat. One of the similarity goal facts was that both the rainwater and the heat are being collected. However, the way the source text phrases the sentences that provide this information resulted in different representations. The text says that the solar collector "absorbs the sunlight" ("solar radiation is absorbed," in the pre-simplified version). Because the representations produced for absorption events are not similar to the representations used for falling events, the fall of rainwater onto the system's collection tray does not align with the absorption of heat by the solar panel.

Wrong Interpretation Choice is similar to SME Representation Mismatch, but refers to situations where the mismatch can be traced to the disambiguation heuristics making an incorrect choice. As noted above, the heuristics used were far from perfect; fully 13.4% of the semantic ambiguities were resolved incorrectly. However, the numbers show that, most of the time, these mismatches did not affect the SME matches.

SME Alignment Mismatch refers to when the base and the target both contain the necessary fact or facts to identify a similarity or difference, but SME did not produce an appropriate correspondence or candidate inference because the match did not align appropriately. An example of this error occurring is when the system was asked to compare the anatomy of the dolphin and the porpoise. One of the goal facts is that dolphins have long noses, while porpoises have flat noses. Because these facts appeared in the same paragraph, LPI produced a base and a target that both included both cases. While it aligned dolphin with porpoise, as constrained by the question asked, it did not align the dolphin's nose in the base with the porpoise's nose in the target. Rather, it aligned the dolphin's nose in the base with the dolphin's nose in the target, and did the same with the porpoise's nose. As a result, no candidate inferences were drawn.

5. Conclusions

The results suggest that Fact-Based Segmentation and Local Paragraph Interpretation are the best of the four methods tested. FBS had the advantage that it produced the most accurate results while achieving a significantly higher generation efficiency than LPI. This comes at the cost of a small amount of additional overhead during the case construction step, as producing cases with FBS

requires extra computation. In the evaluation, LPI had very little trouble with Facts Not In Case errors. This is because the goal facts were all local. In a comparison where facts are spread more evenly across a source text, it would be less effective.

That Local Sentence Interpretation fared the worst is not surprising. While it produced relatively little noise, as indicated by its high generation efficiency, information is spread across multiple sentences too frequently for it to get the information required to make broader comparisons between two entities. Even in situations where smaller cases are desirable, Sentence-Based Segmentation produced much better results with only moderately larger cases.

There appears to be a tradeoff between accuracy and generation efficiency. This result is reasonable; methods that produce more facts are less likely to produce the Facts Not In Case error, provided that the additional facts being added are at least potentially useful. There are instances where including additional facts in the case produced SME Alignment Mismatch errors, but at the case sizes produced by the four methods described here, those are rarer.

6. Related Work

Case-based reasoning (Schank & Cleary, 1994) involves using existing cases to solve problems and answer questions. Such systems tend to use domain-specific retrieval and matching systems, unlike our use of a general-purpose analogical matcher, SME.

Compare&Contrast (Liu, Wagner, & Birnbaum, 2007) uses the web as a source to find cases similar to a seed case. Rather than parsing the entire source, it builds vectors of named and non-named entities to represent the contents. Such feature-based representations cannot support the kinds of explanation generation that we can, given our use of relational representations.

Our approach to learning by reading focuses more on producing comprehensive structured representations than some other systems (at the cost of additional computational overhead). DART (Clark & Harrison, 2009), NELL (Carlson et al., 2010), and KNEXT (Van Durme & Schubert, 2009) represent other efforts in knowledge extraction, producing logical forms. These systems handle a broader range of syntax than the system described in this paper, but the representations produced are simpler. The PRISMATIC knowledge base, used in IBM's Watson project (Fan et al., 2012), uses a simpler representation strategy, treating words themselves as predicates. This is good for factoid question answering, but less so for reasoning tasks. West et al. (2014) describe a system for targeting the web with specific queries in order to extend Freebase (Bollacker et al., 2008), filling in certain types of missing knowledge. All of these systems use a less expressive representational vocabulary than our combination of DRT and Cyc provide.

KA (Peterson et al., 1994) is a proposed system that resembles ours in that it would construct cases from texts and compare them to other cases. This would allow the system to diagnose errors in the design of physical systems. The work described here aims to be more general, and is not tied to any particular domain.

Textual CBR systems have generally focused on building cases from text resources with the goal of using them as pointers to those text resources, rather than building formally represented cases that can be used for reasoning, as the work we describe here does. Generally this has involved minimal NLP. Brüninghaus and Ashley (2001) describe SMILE, which uses methods for using NLP in the law domain to build more sophisticated cases that can be more accurately compared to each other, compared to methods that use bag-of-words techniques, but relies on a human to identify which features are important. Gupta and Aha (2004) describe FACIT, a TCBR system that uses logical forms as its representations. Like most TCBR systems, it operates by

using the generated cases to index complete texts, rather than reasoning directly over the cases produced.

Analogical Dialogue Act classification (Barbella & Forbus, 2011) has been used to construct cases from textual analogies by classifying sentences based on their role in establishing or extending an analogy, and then using those classifications to determine what statements are part of the base and target of the analogy. Like the methods described in this paper, it makes use of connectivity properties of semantic interpretations.

7. Future Work

One of the properties of the fact-based segmentation method is that it depends on coreference resolution, an area where our language system could use improvements. Currently fact-based segmentation uses common collection membership as a method for picking up on topic similarity even where entities are not coreferent; exploring additional means of handling this is one area where the algorithm could be further refined. We also plan to explore whether further improvement is possible by synthesizing our two connection-based methods into a single method.

Another possible avenue for exploiting the cases produced is to combine them with analogical retrieval (Forbus et al., 1997; Forbus, Gentner, & Law, 1995) to answer other types of questions via analogy (e.g. Klenk & Forbus, 2009). Learning more general models of concepts via analogical generalization (McClure & Forbus, 2012) would also be another way to use cases constructed via these methods.

We are currently extending the results in Barbella and Forbus (2011) to make use of a variant of fact-based segmentation as part of building cases that represent the base and the target of an explicitly stated analogy. This requires recognizing certain facts as ones which introduce correspondences between the base and the target, and handling those as exceptions.

Currently, the system produces cases for every possible seed after it reads. It can also produce a set of cases for a particular seed that is targeted. One potential extension for the system is to identify which entities are likely to be the most useful seeds for cases. For example, the entity that names the topic of a paragraph may be more useful as a seed than an arbitrary entity from later in that paragraph.

Acknowledgements

This research was supported by a grant from the Intelligent and Autonomous Systems Program of the Office of Naval Research.

References

Allen, J. F. (1994). *Natural language understanding (2nd Ed.)* Redwood City, CA: Benjamin/Cummings.

Barbella, D., & Forbus, K. (2011). Analogical dialogue acts: Supporting learning by reading analogies in instructional texts. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (pp. 1429-1435). San Francisco, CA: AAAI Press.

Barker, K., Agashe, B., Chaw, S., Fan, J., Glass, M., Hobbs, J., Hovey, E., Israel, D., Kim, D., Mulkar, R., Patwardhan, S., Porter, B., Tecuci, D., & Yeh, P. (2007). Learning by reading: A prototype system, performance baseline, and lessons learned. *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence* (pp. 280-286). Vancouver, BC: AAAI Press.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (pp. 1247-1250). ACM.

Brüninghaus, S., & Ashley, K. D. (2001). The role of information extraction for textual CBR. *Case-based reasoning research and development* (pp. 74-89). Springer Berlin Heidelberg.

Buckley, S. 1979. *Sun Up to Sun Down*. New York, NY: McGraw-Hill.

Carlson, A., Betteridge, B., Kisiel, B., Settles, E. R., Hruschka, E. R., & Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 1306-1313). Atlanta, GA: AAAI Press.

Clark, P., & Harrison, P. (2009). Large-scale extraction and use of knowledge from text. *Proceedings of the Fifth International Conference on Knowledge Capture* (pp. 153-160). Redondo Beach, CA: ACM.

Chaudhri, V., Heymans, S., Spaulding, A., Overholtzer, A., & Wessel, M. (2014). Large-scale analogical reasoning. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (pp. 359-365). Québec City, QC: AAAI Press.

Dolphin vs Porpoise. (n.d.). Retrieved February 9, 2015, from http://www.diffen.com/difference/Dolphin_vs_PorpoiseFalkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41*, 1-63.

Fan, J., Kalyanpur, A., Gondek, D.C., & Ferrucci, D.A. (2012). Automatic knowledge extraction from documents. *IBM Journal of Research & Development, 56*, 5:1-5:10.

Forbus, K., Gentner, D., Everett, J. O., & Wu, M. (1997). Towards a computational model of evaluating and using analogical inferences. *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 229-234).

Forbus, K., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19*, 141-205.

Forbus, K, Klenk, M., & Hinrichs, T. (2009). Companion Cognitive Systems: Design goals and lessons learned so far. *IEEE Intelligent Systems, 24*, 36-46.

Forbus, K., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. (2007). Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by leading. *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence* (pp. 1542-1547). Vancouver, BC: AAAI Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*, 155-170.

Gentner, D., & Gunn, V. (2001). Structural alignment facilitates the noticing of differences. *Memory and Cognition*, 29(4), 565-577.

Grishman, R., Macleod, C., & Wolff, S. (1993). *The COMLEX Syntax Project*. Ft. Belvoir: Defense Technical Information Center.

Gupta, K. M., & Aha, D. W. (2004). Towards acquiring case indexing taxonomies from text. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference* (pp. 172-177). Clearwater Beach, FL: AAAI Press.

Kamp, H., & Reyle, U. (1993). *From discourse to logic: Introduction to model-theoretic semantics of natural language*. Boston, MA: Kluwer Academic.

Klenk, M., & Forbus, K. D. (2009). Domain transfer via cross-domain analogy. *Cognitive Systems Research, Special Issue on Analogies: Integrating Cognitive Abilities*, 10(3), 240-250.

Kuehne, S., & Forbus, K. (2004). Capturing QP-relevant information from natural language text. *Proceedings of the Eighteenth International Qualitative Reasoning Workshop*. Evanston, IL.

Liu, J., Wagner, E., & Birnbaum, L. (2007). Compare&Contrast: Using the web to discover comparable cases for news stories. *Proceedings of the Sixteenth International World Wide Web Conference* (pp. 541-551). Banff, AB.

Macleod, C., Grisham, R., & Meyers, A. (1998). *COMLEX syntax reference manual, Version 3.0*.

McLure, M., & Forbus, K. (2012). Encoding strategies for learning geographical concepts via analogy. *Proceedings of the Twenty-Sixth International Workshop on Qualitative Reasoning*. Los Angeles, CA.

Mostek, T., Forbus, K. D., & Meverden, C. (2000). Dynamic case creation and expansion for analogical reasoning. *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (pp. 323-329). Austin, TX: AAAI Press.

Peterson, J., Mahesh, K., & Goel, A. (1994). Situating natural language understanding within experience-based design. *International journal of human-computer studies*, 41(6), 881-913.

Schank, R., & Cleary, C. (1994). *Engines for Education*. Erlbaum.

Tomai, E., & Forbus, K. (2009). EA NLU: Practical language understanding for cognitive modeling. *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference* (pp. 117-122). Sanibel Island, FL: AAAI Press.

Van Durme, B., & Schubert, L. (2008). Open knowledge extraction through compositional language processing. *Symposium on Semantics in Systems for Text Processing* (pp. 239-254). Stroudsburg, PA: Association for Computational Linguistics.

West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., & Lin, D. (2014). Knowledge base completion via search-based question answering. *Proceedings of the 23rd international conference on World wide web* (pp. 515-526). International World Wide Web Conferences Steering Committee.

Yan, J., Forbus, K., & Gentner, D. (2003). A theory of rerepresentation in analogical matching. *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society* (pp. 1265-1270). Mahwah, NJ: Lawrence Erlbaum.