Journal of Artificial Intelligence Research 33 (2008) 615-655 Submitted 09/08; published 12/08

The Latent Relation Mapping Engine: Algorithm and Experiments

Peter D. Turney PETER.TURNEY@NRC-CNRC.GC.CA
Institute for Information Technology
National Research Council Canada
Ottawa, Ontario, Canada, K1A 0R6

Abstract

Many AI researchers and cognitive scientists have argued that analogy is the core of cognition. The most influential work on computational modeling of analogy-making is Structure Mapping Theory (SMT) and its implementation in the Structure Mapping Engine (SME). A limitation of SME is the requirement for complex hand-coded representations. We introduce the Latent Relation Mapping Engine (LRME), which combines ideas from SME and Latent Relational Analysis (LRA) in order to remove the requirement for hand-coded representations. LRME builds analogical mappings between lists of words, using a large corpus of raw text to automatically discover the semantic relations among the words. We evaluate LRME on a set of twenty analogical mapping problems, ten based on scientific analogies and ten based on common metaphors. LRME achieves human-level performance on the twenty problems. We compare LRME with a variety of alternative approaches and find that they are not able to reach the same level of performance.

1. Introduction

When we are faced with a problem, we try to recall similar problems that we have faced in the past, so that we can transfer our knowledge from past experience to the current problem. We make an analogy between the past situation and the current situation, and we use the analogy to transfer knowledge (Gentner, 1983; Minsky, 1986; Holyoak & Thagard, 1995; Hofstadter, 2001; Hawkins & Blakeslee, 2004).

In his survey of the computational modeling of analogy-making, French (2002) cites Structure Mapping Theory (SMT) (Gentner, 1983) and its implementation in the Structure Mapping Engine (SME) (Falkenhainer, Forbus, & Gentner, 1989) as the most influential work on modeling of analogy-making. In SME, an analogical mapping $M : A \rightarrow B$ is from a source A to a target B. The source is more familiar, more known, or more concrete, whereas the target is relatively unfamiliar, unknown, or abstract. The analogical mapping is used to transfer knowledge from the source to the target.

Gentner (1983) argues that there are two kinds of similarity, attributional similarity and relational similarity. The distinction between attributes and relations may be understood in terms of predicate logic. An attribute is a predicate with one argument, such as $\text{LARGE}(X)$, meaning X is large. A relation is a predicate with two or more arguments, such as $\text{COLLIDES\_WITH}(X, Y)$, meaning X collides with Y.

The Structure Mapping Engine prefers mappings based on relational similarity over mappings based on attributional similarity (Falkenhainer et al., 1989). For example, SME is able to build a mapping from a representation of the solar system (the source) to a representation of the Rutherford-Bohr model of the atom (the target). The sun is mapped to the nucleus, planets are mapped to electrons, and mass is mapped to charge. Note that this mapping emphasizes relational similarity. The sun and the nucleus are very different in terms of their attributes: the sun is very large and the nucleus is very small. Likewise, planets and electrons have little attributional similarity. On the other hand, planets revolve around the sun like electrons revolve around the nucleus. The mass of the sun attracts the mass of the planets like the charge of the nucleus attracts the charge of the electrons.

Gentner (1991) provides evidence that children rely primarily on attributional similarity for mapping, gradually switching over to relational similarity as they mature. She uses the terms *mere appearance* to refer to mapping based mostly on attributional similarity, *analogy* to refer to mapping based mostly on relational similarity, and *literal similarity* to refer to a mixture of attributional and relational similarity. Since we use analogical mappings to solve problems and make predictions, we should focus on structure, especially causal relations, and look beyond the surface attributes of things (Gentner, 1983). The analogy between the solar system and the Rutherford-Bohr model of the atom illustrates the importance of going beyond mere appearance, to the underlying structures.

Figures 1 and 2 show the LISP representations used by SME as input for the analogy between the solar system and the atom (Falkenhainer et al., 1989). Chalmers, French, and Hofstadter (1992) criticize SME's requirement for complex hand-coded representations. They argue that most of the hard work is done by the human who creates these high-level hand-coded representations, rather than by SME.

```
(defEntity sun :type inanimate)
(defEntity planet :type inanimate)

(defDescription solar-system
     entities (sun planet)
     expressions (((mass sun) :name mass-sun)
          ((mass planet) :name mass-planet)
          ((greater mass-sun mass-planet) :name >mass)
          ((attracts sun planet) :name attracts-form)
          ((revolve-around planet sun) :name revolve)
          ((and >mass attracts-form) :name and1)
          ((cause and1 revolve) :name cause-revolve)
          ((temperature sun) :name temp-sun)
          ((temperature planet) :name temp-planet)
          ((greater temp-sun temp-planet) :name >temp)
          ((gravity mass-sun mass-planet) :name force-gravity)
          ((cause force-gravity attracts-form) :name why-attracts)))
```

Figure 1: The representation of the solar system in SME (Falkenhainer et al., 1989).

Gentner, Forbus, and their colleagues have attempted to avoid hand-coding in their recent work with SME.[1] The CogSketch system can generate LISP representations from simple sketches (Forbus, Usher, Lovett, Lockwood, & Wetzel, 2008). The Gizmo system can generate LISP representations from qualitative physics models (Yan & Forbus, 2005). The Learning Reader system can generate LISP representations from natural language text (Forbus et al., 2007). These systems do not require LISP input.

1. Dedre Gentner, personal communication, October 29, 2008.

```
(defEntity nucleus :type inanimate)
(defEntity electron :type inanimate)

(defDescription rutherford-atom
     entities (nucleus electron)
     expressions (((mass nucleus) :name mass-n)
         ((mass electron) :name mass-e)
         ((greater mass-n mass-e) :name >mass)
         ((attracts nucleus electron) :name attracts-form)
         ((revolve-around electron nucleus) :name revolve)
         ((charge electron) :name q-electron)
         ((charge nucleus) :name q-nucleus)
         ((opposite-sign q-nucleus q-electron) :name >charge)
         ((cause >charge attracts-form) :name why-attracts)))
```

Figure 2: The Rutherford-Bohr model of the atom in SME (Falkenhainer et al., 1989).

However, the CogSketch user interface requires the person who draws the sketch to identify the basic components in the sketch and hand-label them with terms from a knowledge base derived from OpenCyc. Forbus et al. (2008) note that OpenCyc contains more than 58,000 hand-coded concepts, and they have added further hand-coded concepts to OpenCyc, in order to support CogSketch. The Gizmo system requires the user to hand-code a physical model, using the methods of qualitative physics (Yan & Forbus, 2005). Learning Reader uses more than 28,000 phrasal patterns, which were derived from ResearchCyc (Forbus et al., 2007). It is evident that SME still requires substantial hand-coded knowledge.

The work we present in this paper is an effort to avoid complex hand-coded representations. Our approach is to combine ideas from SME (Falkenhainer et al., 1989) and Latent Relational Analysis (LRA) (Turney, 2006). We call the resulting algorithm the Latent Relation Mapping Engine (LRME). We represent the semantic relation between two terms using a vector, in which the elements are derived from pattern frequencies in a large corpus of raw text. Because the semantic relations are automatically derived from a corpus, LRME does not require hand-coded representations of relations. It only needs a list of terms from the source and a list of terms from the target. Given these two lists, LRME uses the corpus to build representations of the relations among the terms, and then it constructs a mapping between the two lists.

Tables 1 and 2 show the input and output of LRME for the analogy between the solar system and the Rutherford-Bohr model of the atom. Although some human effort is involved in constructing the input lists, it is considerably less effort than SME requires for its input (contrast Figures 1 and 2 with Table 1).

Scientific analogies, such as the analogy between the solar system and the Rutherford-Bohr model of the atom, may seem esoteric, but we believe analogy-making is ubiquitous in our daily lives. A potential practical application for this work is the task of identifying semantic roles (Gildea & Jurafsky, 2002). Since roles are relations, not attributes, it is appropriate to treat semantic role labeling as an analogical mapping problem.

For example, the Judgement semantic frame contains semantic roles such as judge, evaluee, and reason, and the Statement frame contains roles such as speaker, addressee, message, topic, and medium (Gildea & Jurafsky, 2002). The task of identifying

Source A	**Target** B
planet	revolves
attracts	atom
revolves	attracts
sun	electromagnetism
gravity	nucleus
solar system	charge
mass	electron

Table 1: The representation of the input in LRME.

Source A	**Mapping** M	**Target** B
solar system	$\rightarrow$	atom
sun	$\rightarrow$	nucleus
planet	$\rightarrow$	electron
mass	$\rightarrow$	charge
attracts	$\rightarrow$	attracts
revolves	$\rightarrow$	revolves
gravity	$\rightarrow$	electromagnetism

Table 2: The representation of the output in LRME.

semantic roles is to automatically label sentences with their roles, as in the following examples (Gildea & Jurafsky, 2002):

- [*Judge* She] **blames** [*Evaluee* the Government] [*Reason* for failing to do enough to help].
- [*Speaker* We] **talked** [*Topic* about the proposal] [*Medium* over the phone].

If we have a training set of labeled sentences and a testing set of unlabeled sentences, then we may view the task of labeling the testing sentences as a problem of creating analogical mappings between the training sentences (sources) and the testing sentences (targets). Table 3 shows how "She blames the Government for failing to do enough to help." might be mapped to "They blame the company for polluting the environment." Once a mapping has been found, we can transfer knowledge, in the form of semantic role labels, from the source to the target.

Source A	**Mapping** M	**Target** B
she	$\rightarrow$	they
blames	$\rightarrow$	blame
government	$\rightarrow$	company
failing	$\rightarrow$	polluting
help	$\rightarrow$	environment

Table 3: Semantic role labeling as analogical mapping.

In Section 2, we briefly discuss the hypotheses behind the design of LRME. We then precisely define the task that is performed by LRME, a specific form of analogical mapping, in Section 3. LRME builds on Latent Relational Analysis (LRA), hence we summarize LRA in Section 4. We discuss potential applications of LRME in Section 5.

To evaluate LRME, we created twenty analogical mapping problems, ten science analogy problems (Holyoak & Thagard, 1995) and ten common metaphor problems (Lakoff & Johnson, 1980). Table 1 is one of the science analogy problems. Our intended solution is given in Table 2. To validate our intended solutions, we gave our colleagues the lists of terms (as in Table 1) and asked them to generate mappings between the lists. Section 6 presents the results of this experiment. Across the twenty problems, the average agreement with our intended solutions (as in Table 2) was 87.6%.

The LRME algorithm is outlined in Section 7, along with its evaluation on the twenty mapping problems. LRME achieves an accuracy of 91.5%. The difference between this performance and the human average of 87.6% is not statistically significant.

Section 8 examines a variety of alternative approaches to the analogy mapping task. The best approach achieves an accuracy of 76.8%, but this approach requires hand-coded part-of-speech tags. This performance is significantly below LRME and human performance.

In Section 9, we discuss some questions that are raised by the results in the preceding sections. Related work is described in Section 10, future work and limitations are considered in Section 11, and we conclude in Section 12.

2. Guiding Hypotheses

In this section, we list some of the assumptions that have guided the design of LRME. The results we present in this paper do not necessarily require these assumptions, but it might be helpful to the reader, to understand the reasoning behind our approach.

1. **Analogies and semantic relations:** Analogies are based on semantic relations (Gentner, 1983). For example, the analogy between the solar system and the Rutherford-Bohr model of the atom is based on the similarity of the semantic relations among the concepts involved in our understanding of the solar system to the semantic relations among the concepts involved in the Rutherford-Bohr model of the atom.
2. **Co-occurrences and semantic relations:** Two terms have an interesting, significant semantic relation if and only if they they tend to co-occur within a relatively small window (e.g., five words) in a relatively large corpus (e.g., 10^{10} words). Having an interesting semantic relation causes co-occurrence and co-occurrence is a reliable indicator of an interesting semantic relation (Firth, 1957).
3. **Meanings and semantic relations:** Meaning has more to do with relations among words than individual words. Individual words tend to be ambiguous and polysemous. By putting two words into a pair, we constrain their possible meanings. By putting words into a sentence, with multiple relations among the words in the sentence, we constrain the possible meanings further. If we focus on word pairs (or tuples), instead of individual words, word sense disambiguation is less problematic. Perhaps a word has no sense apart from its relations with other words (Kilgarriff, 1997).
4. **Pattern distributions and semantic relations:** There is a many-to-many mapping between semantic relations and the patterns in which two terms co-occur. For example, the relation CauseEffect(X, Y) may be expressed as "X causes Y", "Y

from X", "Y due to X", "Y because of X", and so on. Likewise, the pattern "Y from X" may be an expression of CauseEffect(X, Y) ("sick from bacteria") or OriginEntity(X, Y) ("oranges from Spain"). However, for a given X and Y, the statistical distribution of patterns in which X and Y co-occur is a reliable signature of the semantic relations between X and Y (Turney, 2006).

To the extent that LRME works, we believe its success lends some support to these hypotheses.

3. The Task

In this paper, we examine algorithms that generate analogical mappings. For simplicity, we restrict the task to generating *bijective* mappings; that is, mappings that are both *injective* (one-to-one; there is no instance in which two terms in the source map to the same term in the target) and *surjective* (onto; the source terms cover all of the target terms; there is no target term that is left out of the mapping). We assume that the entities that are to be mapped are given as input. Formally, the input I for the algorithms is two sets of terms, A and B.

$$I = \{\langle A, B\rangle\} \tag{1}$$

Since the mappings are bijective, A and B must contain the same number of terms, m.

$$A = \{a_1, a_2, \ldots, a_m\} \tag{2}$$
$$B = \{b_1, b_2, \ldots, b_m\} \tag{3}$$

A term, a_i or b_j, may consist of a single word (*planet*) or a compound of two or more words (*solar system*). The words may be any part of speech (nouns, verbs, adjectives, or adverbs). The output O is a bijective mapping M from A to B.

$$O = \{M : A \rightarrow B\} \tag{4}$$
$$M(a_i) \in B \tag{5}$$
$$M(A) = \{M(a_1), M(a_2), \ldots, M(a_m)\} = B \tag{6}$$

The algorithms that we consider here can accept a batch of multiple independent mapping problems as input and generate a mapping for each one as output.

$$I = \{\langle A_1, B_1\rangle, \langle A_2, B_2\rangle, \ldots, \langle A_n, B_n\rangle\} \tag{7}$$
$$O = \{M_1 : A_1 \rightarrow B_1, M_2 : A_2 \rightarrow B_2, \ldots, M_n : A_n \rightarrow B_n\} \tag{8}$$

Suppose the terms in A are in some arbitrary order $\mathbf{a}$.

$$\mathbf{a} = \langle a_1, a_2, \ldots, a_m\rangle \tag{9}$$

The mapping function $M : A \rightarrow B$, given $\mathbf{a}$, determines a unique ordering $\mathbf{b}$ of B.

$$\mathbf{b} = \langle M(a_1), M(a_2), \ldots, M(a_m) \rangle \tag{10}$$

Likewise, an ordering $\mathbf{b}$ of B, given $\mathbf{a}$, defines a unique mapping function M. Since there are $m!$ possible orderings of B, there are also $m!$ possible mappings from A to B. The task is to search through the $m!$ mappings and find the best one. (Section 6 shows that there is a relatively high degree of consensus about which mappings are best.)

Let $P(A, B)$ be the set of all $m!$ bijective mappings from A to B. (P stands for *permutation*, since each mapping corresponds to a permutation.)

$$P(A, B) = \{M_1, M_2, \ldots, M_{m!}\} \tag{11}$$

$$m = |A| = |B| \tag{12}$$

$$m! = |P(A, B)| \tag{13}$$

In the following experiments, m is 7 on average and 9 at most, so $m!$ is usually around $7! = 5{,}040$ and at most $9! = 362{,}880$. It is feasible for us to exhaustively search $P(A, B)$.

We explore two basic kinds of algorithms for generating analogical mappings, algorithms based on *attributional similarity* and algorithms based on *relational similarity* (Turney, 2006). The attributional similarity between two words, $\mathrm{sim_a}(a, b) \in \Re$, depends on the degree of correspondence between the properties of a and b. The more correspondence there is, the greater their attributional similarity. The relational similarity between two *pairs* of words, $\mathrm{sim_r}(a:b, c:d) \in \Re$, depends on the degree of correspondence between the relations of $a:b$ and $c:d$. The more correspondence there is, the greater their relational similarity. For example, *dog* and *wolf* have a relatively high degree of attributional similarity, whereas *dog*: *bark* and *cat*: *meow* have a relatively high degree of relational similarity.

Attributional mapping algorithms seek the mapping (or mappings) M_a that maximizes the sum of the attributional similarities between the terms in A and the corresponding terms in B. (When there are multiple mappings that maximize the sum, we break the tie by randomly choosing one of them.)

$$M_\mathrm{a} = \underset{M \in P(A,B)}{\arg\max} \sum_{i=1}^{m} \mathrm{sim_a}(a_i, M(a_i)) \tag{14}$$

Relational mapping algorithms seek the mapping (or mappings) M_r that maximizes the sum of the relational similarities.

$$M_\mathrm{r} = \underset{M \in P(A,B)}{\arg\max} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mathrm{sim_r}(a_i : a_j, M(a_i) : M(a_j)) \tag{15}$$

In (15), we assume that $\mathrm{sim_r}$ is symmetrical. For example, the degree of relational similarity between *dog*: *bark* and *cat*: *meow* is the same as the degree of relational similarity between *bark*: *dog* and *meow*: *cat*.

$$\mathrm{sim_r}(a:b, c:d) = \mathrm{sim_r}(b:a, d:c) \tag{16}$$

We also assume that $\mathrm{sim_r}(a:a, b:b)$ is not interesting; for example, it may be some constant value for all a and b. Therefore (15) is designed so that i is always less than j.

Let $\mathrm{score}_\mathrm{r}(M)$ and $\mathrm{score}_\mathrm{a}(M)$ be defined as follows.

$$\mathrm{score}_\mathrm{r}(M) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mathrm{sim}_\mathrm{r}(a_i \!:\! a_j, M(a_i) \!:\! M(a_j)) \tag{17}$$

$$\mathrm{score}_\mathrm{a}(M) = \sum_{i=1}^{m} \mathrm{sim}_\mathrm{a}(a_i, M(a_i)) \tag{18}$$

Now M_r and M_a may be defined in terms of $\mathrm{score}_\mathrm{r}(M)$ and $\mathrm{score}_\mathrm{a}(M)$.

$$M_\mathrm{r} = \operatorname*{arg\,max}_{M \in P(A,B)} \mathrm{score}_\mathrm{r}(M) \tag{19}$$

$$M_\mathrm{a} = \operatorname*{arg\,max}_{M \in P(A,B)} \mathrm{score}_\mathrm{a}(M) \tag{20}$$

M_r is the best mapping according to sim_r and M_a is the best mapping according to sim_a.

Recall Gentner's (1991) terms, discussed in Section 1, *mere appearance* (mostly attributional similarity), *analogy* (mostly relational similarity), and *literal similarity* (a mixture of attributional and relational similarity). We take it that M_r is an abstract model of mapping based on analogy and M_a is a model of mere appearance. For literal similarity, we can combine M_r and M_a, but we should take care to normalize $\mathrm{score}_\mathrm{r}(M)$ and $\mathrm{score}_\mathrm{a}(M)$ before we combine them. (We experiment with combining them in Section 9.2.)

4. Latent Relational Analysis

LRME uses a simplified form of Latent Relational Analysis (LRA) (Turney, 2005, 2006) to calculate the relational similarity between pairs of words. We will briefly describe past work with LRA before we present LRME.

LRA takes as input I a set of word pairs and generates as output O the relational similarity $\mathrm{sim}_\mathrm{r}(a_i \!:\! b_i, a_j \!:\! b_j)$ between any two pairs in the input.

$$I = \{a_1 \!:\! b_1, a_2 \!:\! b_2, \ldots, a_n \!:\! b_n\} \tag{21}$$

$$O = \{\mathrm{sim}_\mathrm{r} : I \times I \rightarrow \Re\} \tag{22}$$

LRA was designed to evaluate proportional analogies. Proportional analogies have the form $a \!:\! b \!::\! c \!:\! d$, which means "$a$ is to b as c is to d". For example, *mason* : *stone* :: *carpenter* : *wood* means "mason is to stone as carpenter is to wood". A mason is an artisan who works with stone and a carpenter is an artisan who works with wood.

We consider proportional analogies to be a special case of bijective analogical mapping, as defined in Section 3, in which $|A| = |B| = m = 2$. For example, $a_1 \!:\! a_2 \!::\! b_1 \!:\! b_2$ is equivalent to M_0 in (23).

$$A = \{a_1, a_2\}\,,\ B = \{b_1, b_2\}\,,\ M_0(a_1) = b_1,\ M_0(a_2) = b_2. \tag{23}$$

From the definition of $\mathrm{score}_\mathrm{r}(M)$ in (17), we have the following result for M_0.

$$\mathrm{score_r}(M_0) = \mathrm{sim_r}(a_1\!:\!a_2, M_0(a_1)\!:\!M_0(a_2)) = \mathrm{sim_r}(a_1\!:\!a_2, b_1\!:\!b_2) \tag{24}$$

That is, the quality of the proportional analogy *mason* : *stone* :: *carpenter* : *wood* is given by $\mathrm{sim_r}(mason\!:\!stone, carpenter\!:\!wood)$.

Proportional analogies may also be evaluated using attributional similarity. From the definition of $\mathrm{score_a}(M)$ in (18), we have the following result for M_0.

$$\mathrm{score_a}(M_0) = \mathrm{sim_a}(a_1, M_0(a_1)) + \mathrm{sim_a}(a_2, M_0(a_2)) = \mathrm{sim_a}(a_1, b_1) + \mathrm{sim_a}(a_2, b_2) \tag{25}$$

For attributional similarity, the quality of the proportional analogy *mason* : *stone* :: *carpenter* : *wood* is given by $\mathrm{sim_a}(mason, carpenter) + \mathrm{sim_a}(stone, wood)$.

LRA only handles proportional analogies. The main contribution of LRME is to extend LRA beyond proportional analogies to bijective analogies for which $m > 2$.

Turney (2006) describes ten potential applications of LRA: recognizing proportional analogies, structure mapping theory, modeling metaphor, classifying semantic relations, word sense disambiguation, information extraction, question answering, automatic thesaurus generation, information retrieval, and identifying semantic roles. Two of these applications (evaluating proportional analogies and classifying semantic relations) are experimentally evaluated, with state-of-the-art results.

Turney (2006) compares the performance of relational similarity (24) and attributional similarity (25) on the task of solving 374 multiple-choice proportional analogy questions from the SAT college entrance test. LRA is used to measure relational similarity and a variety of lexicon-based and corpus-based algorithms are used to measure attributional similarity. LRA achieves an accuracy of 56% on the 374 SAT questions, which is not significantly different from the average human score of 57%. On the other hand, the best performance by attributional similarity is 35%. The results show that attributional similarity is better than random guessing, but not as good as relational similarity. This result is consistent with Gentner's (1991) theory of the maturation of human similarity judgments.

Turney (2006) also applies LRA to the task of classifying semantic relations in noun-modifier expressions. A noun-modifier expression is a phrase, such as *laser printer*, in which the head noun (*printer*) is preceded by a modifier (*laser*). The task is to identify the semantic relation between the noun and the modifier. In this case, the relation is *instrument*; the laser is an *instrument* used by the printer. On a set of 600 hand-labeled noun-modifier pairs with five different classes of semantic relations, LRA attains 58% accuracy.

Turney (2008) employs a variation of LRA for solving four different language tests, achieving 52% accuracy on SAT analogy questions, 76% accuracy on TOEFL synonym questions, 75% accuracy on the task of distinguishing synonyms from antonyms, and 77% accuracy on the task of distinguishing words that are similar, words that are associated, and words that are both similar and associated. The same core algorithm is used for all four tests, with no tuning of the parameters to the particular test.

5. Applications for LRME

Since LRME is an extension of LRA, every potential application of LRA is also a potential application of LRME. The advantage of LRME over LRA is the ability to handle bijective analogies when $m > 2$ (where $m = |A| = |B|$). In this section, we consider the kinds of applications that might benefit from this ability.

In Section 7.2, we evaluate LRME on science analogies and common metaphors, which supports the claim that these two applications benefit from the ability to handle larger sets of terms. In Section 1, we saw that identifying semantic roles (Gildea & Jurafsky, 2002) also involves more than two terms, and we believe that LRME will be superior to LRA for semantic role labeling.

Semantic relation classification usually assumes that the relations are binary; that is, a semantic relation is a connection between two terms (Rosario & Hearst, 2001; Nastase & Szpakowicz, 2003; Turney, 2006; Girju et al., 2007). Yuret observed that binary relations may be linked by underlying n-ary relations.[2] For example, Nastase and Szpakowicz (2003) defined a taxonomy of 30 binary semantic relations. Table 4 shows how six binary relations from Nastase and Szpakowicz (2003) can be covered by one 5-ary relation, Agent:Tool:Action:Affected:Theme. An Agent uses a Tool to perform an Action. Somebody or something is Affected by the Action. The whole event can be summarized by its Theme.

Nastase and Szpakowicz (2003)		
Relation	**Example**	**Agent:Tool:Action:Affected:Theme**
agent	student protest	Agent:Action
purpose	concert hall	Theme:Tool
beneficiary	student discount	Affected:Action
instrument	laser printer	Tool:Agent
object	metal separator	Affected:Tool
object property	sunken ship	Action:Affected

Table 4: How six binary semantic relations from Nastase and Szpakowicz (2003) can be viewed as different fragments of one 5-ary semantic relation.

In SemEval Task 4, we found it easier to manually tag the datasets when we expanded binary relations to their underlying n-ary relations (Girju et al., 2007). We believe that this expansion would also facilitate automatic classification of semantic relations. The results in Section 9.3 suggest that all of the applications for LRA that we discussed in Section 4 might benefit from being able to handle bijective analogies when $m > 2$.

6. The Mapping Problems

To evaluate our algorithms for analogical mapping, we created twenty mapping problems, given in Appendix A. The twenty problems consist of ten science analogy problems, based on examples of analogy in science from Chapter 8 of Holyoak and Thagard (1995), and ten common metaphor problems, derived from Lakoff and Johnson (1980).

The tables in Appendix A show our intended mappings for each of the twenty problems. To validate these mappings, we invited our colleagues in the Institute for Information Technology to participate in an experiment. The experiment was hosted on a web server

2. Deniz Yuret, personal communication, February 13, 2007. This observation was in the context of our work on building the datasets for SemEval 2007 Task 4 (Girju et al., 2007).

(only accessible inside our institute) and people participated anonymously, using their web browsers in their offices. There were 39 volunteers who began the experiment and 22 who went all the way to the end. In our analysis, we use only the data from the 22 participants who completed all of the mapping problems.

The instructions for the participants are in Appendix A. The sequence of the problems and the order of the terms within a problem were randomized separately for each participant, to remove any effects due to order. Table 5 shows the agreement between our intended mapping and the mappings generated by the participants. Across the twenty problems, the average agreement was 87.6%, which is higher than the agreement figures for many linguistic annotation tasks. This agreement is impressive, given that the participants had minimal instructions and no training.

Type	**Mapping**	**Source → Target**	**Agreement**	m
	A1	solar system → atom	90.9	7
	A2	water flow → heat transfer	86.9	8
	A3	waves → sounds	81.8	8
	A4	combustion → respiration	79.0	8
science	A5	sound → light	79.2	7
analogies	A6	projectile → planet	97.4	7
	A7	artificial selection → natural selection	74.7	7
	A8	billiard balls → gas molecules	88.1	8
	A9	computer → mind	84.3	9
	A10	slot machine → bacterial mutation	83.6	5
	M1	war → argument	93.5	7
	M2	buying an item → accepting a belief	96.1	7
	M3	grounds for a building → reasons for a theory	87.9	6
	M4	impediments to travel → difficulties	100.0	7
common	M5	money → time	77.3	6
metaphors	M6	seeds → ideas	89.0	7
	M7	machine → mind	98.7	7
	M8	object → idea	89.1	5
	M9	following → understanding	96.6	8
	M10	seeing → understanding	78.8	6
Average			87.6	7.0

Table 5: The average agreement between our intended mappings and the mappings of the 22 participants. See Appendix A for the details.

The column labeled m gives the number of terms in the set of source terms for each mapping problem (which is equal to the number of terms in the set of target terms). For the average problem, $m = 7$. The third column in Table 5 gives a mnemonic that summarizes the mapping (e.g., solar system → atom). Note that the mnemonic is not used as input for any of the algorithms, nor was the mnemonic shown to the participants in the experiment.

The agreement figures in Table 5 for each individual mapping problem are averages over the m mappings for each problem. Appendix A gives a more detailed view, showing the agreement for each individual mapping in the m mappings. The twenty problems contain a total of 140 individual mappings (20×7). Appendix A shows that every one of these 140 mappings has an agreement of 50% or higher. That is, in every case, the majority of the participants agreed with our intended mapping. (There are two cases where the agreement is exactly 50%. See problems A5 in Table 14 and M5 in Table 16 in Appendix A.)

If we select the mapping that is chosen by the majority of the 22 participants, then we will get a perfect score on all twenty problems. More precisely, if we try all $m!$ mappings for each problem, and select the mapping that maximizes the sum of the number of participants who agree with each individual mapping in the m mappings, then we will have a score of 100% on all twenty problems. This is strong support for the intended mappings that are given in Appendix A.

In Section 3, we applied Genter's (1991) categories – *mere appearance* (mostly attributional similarity), *analogy* (mostly relational similarity), and *literal similarity* (a mixture of attributional and relational similarity) – to the mappings M_{r} and M_{a}, where M_{r} is the best mapping according to $\mathrm{sim}_{\mathrm{r}}$ and M_{a} is the best mapping according to $\mathrm{sim}_{\mathrm{a}}$. The twenty mapping problems were chosen as analogy problems; that is, the intended mappings in Appendix A are meant to be relational mappings, M_{r}; mappings that maximize relational similarity, $\mathrm{sim}_{\mathrm{r}}$. We have tried to avoid mere appearance and literal similarity.

In Section 7 we use the twenty mapping problems to evaluate a relational mapping algorithm (LRME), and in Section 8 we use them to evaluate several different attributional mapping algorithms. Our hypothesis is that LRME will perform significantly better than any of the attributional mapping algorithms on the twenty mapping problems, because they are analogy problems (not mere appearance problems and not literal similarity problems). We expect relational and attributional mapping algorithms would perform approximately equally well on literal similarity problems, and we expect that mere appearance problems would favour attributional algorithms over relational algorithms, but we do not test these latter two hypotheses, because our primary interest in this paper is analogy-making.

Our goal is to test the hypothesis that there is a real, practical, effective, measurable difference between the output of LRME and the output of the various attributional mapping algorithms. A skeptic might claim that relational similarity $\mathrm{sim}_{\mathrm{r}}(a : b, c : d)$ can be reduced to attributional similarity $\mathrm{sim}_{\mathrm{a}}(a, c) + \mathrm{sim}_{\mathrm{a}}(b, d)$; therefore our relational mapping algorithm is a complicated solution to an illusory problem. A slightly less skeptical claim is that relational similarity versus attributional similarity is a valid distinction in cognitive psychology, but our relational mapping algorithm does not capture this distinction. To test our hypothesis and refute these skeptical claims, we have created twenty analogical mapping problems, and we will show that LRME handles these problems significantly better than the various attributional mapping algorithms.

7. The Latent Relation Mapping Engine

The Latent Relation Mapping Engine (LRME) seeks the mapping M_{r} that maximizes the sum of the relational similarities.

$$M_{\mathrm{r}} = \operatorname*{arg\,max}_{M \in P(A,B)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mathrm{sim}_{\mathrm{r}}(a_i\!:\!a_j, M(a_i)\!:\!M(a_j)) \tag{26}$$

We search for M_{r} by exhaustively evaluating all of the possibilities. Ties are broken randomly. We use a simplified form of LRA (Turney, 2006) to calculate $\mathrm{sim}_{\mathrm{r}}$.

7.1 Algorithm

Briefly, the idea of LRME is to build a pair-pattern matrix $\mathbf{X}$, in which the rows correspond to pairs of terms and the columns correspond to patterns. For example, the row $\mathbf{x}_{i:}$ might correspond to the pair of terms *sun* : *solar system* and the column $\mathbf{x}_{:j}$ might correspond to the pattern "$*$ X centered Y $*$". In these patterns, "$*$" is a wild card, which can match any single word. The value of an element x_{ij} in $\mathbf{X}$ is based on the frequency of the pattern for $\mathbf{x}_{:j}$, when X and Y are instantiated by the terms in the pair for $\mathbf{x}_{i:}$. For example, if we take the pattern "$*$ X centered Y $*$" and instantiate $X : Y$ with the pair *sun* : *solar system*, then we have the pattern "$*$ sun centered solar system $*$", and thus the value of the element x_{ij} is based on the frequency of "$*$ sun centered solar system $*$" in the corpus. The matrix $\mathbf{X}$ is smoothed with a truncated singular value decomposition (SVD) (Golub & Van Loan, 1996) and the relational similarity $\mathrm{sim_r}$ between two pairs of terms is given by the cosine of the angle between the two corresponding row vectors in $\mathbf{X}$.

In more detail, LRME takes as input I a set of mapping problems and generates as output O a corresponding set of mappings.

$$I = \{\langle A_1, B_1\rangle, \langle A_2, B_2\rangle, \ldots, \langle A_n, B_n\rangle\} \tag{27}$$
$$O = \{M_1 : A_1 \rightarrow B_1, M_2 : A_2 \rightarrow B_2, \ldots, M_n : A_n \rightarrow B_n\} \tag{28}$$

In the following experiments, all twenty mapping problems (Appendix A) are processed in one batch ($n = 20$).

The first step is to make a list R that contains all pairs of terms in the input I. For each mapping problem $\langle A, B\rangle$ in I, we add to R all pairs $a_i : a_j$, such that a_i and a_j are members of A, $i \neq j$, and all pairs $b_i : b_j$, such that b_i and b_j are members of B, $i \neq j$. If $|A| = |B| = m$, then there are $m(m-1)$ pairs from A and $m(m-1)$ pairs from B.[3] A typical pair in R would be *sun* : *solar system*. We do not allow duplicates in R; R is a list of pair types, not pair tokens. For our twenty mapping problems, R is a list of 1,694 pairs.

For each pair r in R, we make a list $S(r)$ of the phrases in the corpus that contain the pair r. Let $a_i : a_j$ be the terms in the pair r. We search in the corpus for all phrases of the following form:

$$\text{“[0 to 1 words] } a_i \text{ [0 to 3 words] } a_j \text{ [0 to 1 words]”} \tag{29}$$

If $a_i : a_j$ is in R, then $a_j : a_i$ is also in R, so we find phrases with the members of the pairs in both orders, $S(a_i : a_j)$ and $S(a_j : a_i)$. The search template (29) is the same as used by Turney (2008).

In the following experiments, we search in a corpus of 5×10^{10} English words (about 280 GB of plain text), consisting of web pages gathered by a web crawler.[4] To retrieve phrases

3. We have $m(m-1)$ here, not $m(m-1)/2$, because we need the pairs in both orders. We only want to calculate $\mathrm{sim_r}$ for one order of the pairs, because i is always less than j in (26); however, to ensure that $\mathrm{sim_r}$ is symmetrical, as in (16), we need to make the matrix $\mathbf{X}$ symmetrical, by having rows in the matrix for both orders of every pair.
4. The corpus was collected by Charles Clarke at the University of Waterloo. We can provide copies of the corpus on request.

from the corpus, we use Wumpus (Büttcher & Clarke, 2005), an efficient search engine for passage retrieval from large corpora.[5]

With the 1,694 pairs in R, we find a total of 1,996,464 phrases in the corpus, an average of about 1,180 phrases per pair. For the pair $r =$ *sun*: *solar system*, a typical phrase s in $S(r)$ would be "a sun centered solar system illustrates".

Next we make a list C of patterns, based on the phrases we have found. For each pair r in R, where $r = a_i : a_j$, if we found a phrase s in $S(r)$, then we replace a_i in s with X and we replace a_j with Y. The remaining words may be either left as they are or replaced with a wild card symbol "$*$". We then replace a_i in s with Y and a_j with X, and replace the remaining words with wild cards or leave them as they are. If there are n remaining words in s, after a_i and a_j are replaced, then we generate 2^{n+1} patterns from s, and we add these patterns to C. We only add new patterns to C; that is, C is a list of pattern types, not pattern tokens; there are no duplicates in C.

For example, for the pair *sun*: *solar system*, we found the phrase "a sun centered solar system illustrates". When we replace $a_i : a_j$ with $X : Y$, we have "a X centered Y illustrates". There are three remaining words, so we can generate eight patterns, such as "a $X * Y$ illustrates", "a X centered $Y *$", "$* X * Y$ illustrates", and so on. Each of these patterns is added to C. Then we replace $a_i : a_j$ with $Y : X$, yielding "a Y centered X illustrates". This gives us another eight patterns, such as "a Y centered $X *$". Thus the phrase "a sun centered solar system illustrates" generates a total of sixteen patterns, which we add to C.

Now we revise R, to make a list of pairs that will correspond to rows in the frequency matrix $\mathbf{F}$. We remove any pairs from R for which no phrases were found in the corpus, when the terms were in either order. Let $a_i : a_j$ be the terms in the pair r. We remove r from R if both $S(a_i : a_j)$ and $S(a_j : a_i)$ are empty. We remove such rows because they would correspond to zero vectors in the matrix $\mathbf{F}$. This reduces R from 1,694 pairs to 1,662 pairs. Let n_r be the number of pairs in R.

Next we revise C, to make a list of patterns that will correspond to columns in the frequency matrix $\mathbf{F}$. In the following experiments, at this stage, C contains millions of patterns, too many for efficient processing with a standard desktop computer. We need to reduce C to a more manageable size. We select the patterns that are shared by the most pairs. Let c be a pattern in C. Let r be a pair in R. If there is a phrase s in $S(r)$, such that there is a pattern generated from s that is identical to c, then we say that r is one of the pairs that generated c. We sort the patterns in C in descending order of the number of pairs in R that generated each pattern, and we select the top tn_r patterns from this sorted list. Following Turney (2008), we set the parameter t to 20; hence C is reduced to the top 33,240 patterns ($tn_r = 20 \times 1{,}662 = 33{,}240$). Let n_c be the number of patterns in C ($n_c = tn_r$).

Now that the rows R and columns C are defined, we can build the frequency matrix $\mathbf{F}$. Let r_i be the i-th pair of terms in R (e.g., let r_i be *sun*: *solar system*) and let c_j be the j-th pattern in C (e.g., let c_j be "$* X$ centered $Y *$"). We instantiate X and Y in the pattern c_j with the terms in r_i ("$*$ sun centered solar system $*$"). The element f_{ij} in $\mathbf{F}$ is the frequency of this instantiated pattern in the corpus.

5. Wumpus was developed by Stefan Büttcher and it is available at http://www.wumpus-search.org/.

Note that we do not need to search again in the corpus for the instantiated pattern for f_{ij}, in order to find its frequency. In the process of creating each pattern, we can keep track of how many phrases generated the pattern, for each pair. We can get the frequency for f_{ij} by checking our record of the patterns that were generated by r_i.

The next step is to transform the matrix $\mathbf{F}$ of raw frequencies into a form $\mathbf{X}$ that enhances the similarity measurement. Turney (2006) used the log entropy transformation, as suggested by Landauer and Dumais (1997). This is a kind of tf-idf (term frequency times inverse document frequency) transformation, which gives more weight to elements in the matrix that are statistically surprising. However, Bullinaria and Levy (2007) recently achieved good results with a new transformation, called PPMIC (Positive Pointwise Mutual Information with Cosine); therefore LRME uses PPMIC. The raw frequencies in $\mathbf{F}$ are used to calculate probabilities, from which we can calculate the pointwise mutual information (PMI) of each element in the matrix. Any element with a negative PMI is then set to zero.

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} f_{ij}} \tag{30}$$

$$p_{i*} = \frac{\sum_{j=1}^{n_c} f_{ij}}{\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} f_{ij}} \tag{31}$$

$$p_{*j} = \frac{\sum_{i=1}^{n_r} f_{ij}}{\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} f_{ij}} \tag{32}$$

$$\mathrm{pmi}_{ij} = \log\left(\frac{p_{ij}}{p_{i*}p_{*j}}\right) \tag{33}$$

$$x_{ij} = \begin{cases} \mathrm{pmi}_{ij} & \text{if } \mathrm{pmi}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

Let r_i be the i-th pair of terms in R (e.g., let r_i be *sun*:*solar system*) and let c_j be the j-th pattern in C (e.g., let c_j be "$*$ X centered Y $*$"). In (33), p_{ij} is the estimated probability of the of the pattern c_j instantiated with the pair r_i ("$*$ sun centered solar system $*$"), p_{i*} is the estimated probability of r_i, and p_{*j} is the estimated probability of c_j. If r_i and c_j are statistically independent, then $p_{i*}p_{*j} = p_{ij}$ (by the definition of independence), and thus pmi_{ij} is zero (since $\log(1) = 0$). If there is an interesting semantic relation between the terms in r_i, and the pattern c_j captures an aspect of that semantic relation, then we should expect p_{ij} to be larger than it would be if r_i and c_j were indepedent; hence we should find that $p_{ij} > p_{i*}p_{*j}$, and thus pmi_{ij} is positive. (See Hypothesis 2 in Section 2.) On the other hand, terms from completely different domains may avoid each other, in which case we should find that pmi_{ij} is negative. PPMIC is designed to give a high value to x_{ij} when the pattern c_j captures an aspect of the semantic relation between the terms in r_i; otherwise, x_{ij} should have a value of zero, indicating that the pattern c_j tells us nothing about the semantic relation between the terms in r_i.

In our experiments, $\mathbf{F}$ has a density of 4.6% (the percentage of nonzero elements) and $\mathbf{X}$ has a density of 3.8%. The lower density of $\mathbf{X}$ is due to elements with a negative PMI, which are transformed to zero by PPMIC.

Now we smooth $\mathbf{X}$ by applying a truncated singular value decomposition (SVD) (Golub & Van Loan, 1996). We use SVDLIBC to calculate the SVD of $\mathbf{X}$.[6] SVDLIBC is designed for sparse (low density) matrices. SVD decomposes $\mathbf{X}$ into the product of three matrices $\mathbf{U\Sigma V}^\mathsf{T}$, where $\mathbf{U}$ and $\mathbf{V}$ are in column orthonormal form (i.e., the columns are orthogonal and have unit length, $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$) and $\mathbf{\Sigma}$ is a diagonal matrix of singular values (Golub & Van Loan, 1996). If $\mathbf{X}$ is of rank r, then $\mathbf{\Sigma}$ is also of rank r. Let $\mathbf{\Sigma}_k$, where $k < r$, be the diagonal matrix formed from the top k singular values, and let $\mathbf{U}_k$ and $\mathbf{V}_k$ be the matrices produced by selecting the corresponding columns from $\mathbf{U}$ and $\mathbf{V}$. The matrix $\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\mathsf{T}$ is the matrix of rank k that best approximates the original matrix $\mathbf{X}$, in the sense that it minimizes the approximation errors. That is, $\hat{\mathbf{X}} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\mathsf{T}$ minimizes $\|\hat{\mathbf{X}} - \mathbf{X}\|_F$ over all matrices $\hat{\mathbf{X}}$ of rank k, where $\|\ldots\|_F$ denotes the Frobenius norm (Golub & Van Loan, 1996). We may think of this matrix $\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\mathsf{T}$ as a smoothed or compressed version of the original matrix $\mathbf{X}$. Following Turney (2006), we set the parameter k to 300.

The relational similarity sim_r between two pairs in R is the inner product of the two corresponding rows in $\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\mathsf{T}$, after the rows have been normalized to unit length. We can simplify calculations by dropping $\mathbf{V}_k$ (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990). We take the matrix $\mathbf{U}_k\mathbf{\Sigma}_k$ and normalize each row to unit length. Let $\mathbf{W}$ be the resulting matrix. Now let $\mathbf{Z}$ be $\mathbf{WW}^\mathsf{T}$, a square matrix of size $n_r \times n_r$. This matrix contains the cosines of all combinations of two pairs in R.

For a mapping problem $\langle A, B\rangle$ in I, let $a : a'$ be a pair of terms from A and let $b : b'$ be a pair of terms from B. Suppose that $r_i = a : a'$ and $r_j = b : b'$, where r_i and r_j are the i-th and j-th pairs in R. Then $\text{sim}_\text{r}(a : a', b : b') = z_{ij}$, where z_{ij} is the element in the i-th row and j-th column of $\mathbf{Z}$. If either $a : a'$ or $b : b'$ is not in R, because $S(a : a')$, $S(a' : a)$, $S(b : b')$, or $S(b' : b)$ is empty, then we set the similarity to zero. Finally, for each mapping problem in I, we output the map M_r that maximizes the sum of the relational similarities.

$$M_\text{r} = \underset{M \in P(A,B)}{\arg\max} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \text{sim}_\text{r}(a_i : a_j, M(a_i) : M(a_j)) \tag{35}$$

The simplified form of LRA used here to calculate sim_r differs from LRA used by Turney (2006) in several ways. In LRME, there is no use of synonyms to generate alternate forms of the pairs of terms. In LRME, there is no morphological processing of the terms. LRME uses PPMIC (Bullinaria & Levy, 2007) to process the raw frequencies, instead of log entropy. Following Turney (2008), LRME uses a slightly different search template (29) and LRME sets the number of columns n_c to tn_r, instead of using a constant. In Section 7.2, we evaluate the impact of two of these changes (PPMIC and n_c), but we have not tested the other changes, which were mainly motivated by a desire for increased efficiency and simplicity.

7.2 Experiments

We implemented LRME in Perl, making external calls to Wumpus for searching the corpus and to SVDLIBC for calculating SVD. We used the Perl Net::Telnet package for interprocess

6. SVDLIBC is the work of Doug Rohde and it is available at http://tedlab.mit.edu/~dr/svdlibc/.

communication with Wumpus, the PDL (Perl Data Language) package for matrix manipulations (e.g., calculating cosines), and the List::Permutor package to generate permutations (i.e., to loop through $P(A, B)$).

We ran the following experiments on a dual core AMD Opteron 64 computer, running 64 bit Linux. Most of the running time is spent searching the corpus for phrases. It took 16 hours and 27 minutes for Wumpus to fetch the 1,996,464 phrases. The remaining steps took 52 minutes, of which SVD took 10 minutes. The running time could be cut in half by using RAID 0 to speed up disk access.

Table 6 shows the performance of LRME in its baseline configuration. For comparison, the agreement of the 22 volunteers with our intended mapping has been copied from Table 5. The difference between the performance of LRME (91.5%) and the human participants (87.6%) is not statistically significant (paired t-test, 95% confidence level).

		Accuracy	
Mapping	**Source → Target**	**LRME**	**Humans**
A1	solar system → atom	100.0	90.9
A2	water flow → heat transfer	100.0	86.9
A3	waves → sounds	100.0	81.8
A4	combustion → respiration	100.0	79.0
A5	sound → light	71.4	79.2
A6	projectile → planet	100.0	97.4
A7	artificial selection → natural selection	71.4	74.7
A8	billiard balls → gas molecules	100.0	88.1
A9	computer → mind	55.6	84.3
A10	slot machine → bacterial mutation	100.0	83.6
M1	war → argument	71.4	93.5
M2	buying an item → accepting a belief	100.0	96.1
M3	grounds for a building → reasons for a theory	100.0	87.9
M4	impediments to travel → difficulties	100.0	100.0
M5	money → time	100.0	77.3
M6	seeds → ideas	100.0	89.0
M7	machine → mind	100.0	98.7
M8	object → idea	60.0	89.1
M9	following → understanding	100.0	96.6
M10	seeing → understanding	100.0	78.8
Average		91.5	87.6

Table 6: LRME in its baseline configuration, compared with human performance.

In Table 6, the column labeled *Humans* is the average of 22 people, whereas the *LRME* column is only one algorithm (it is not an average). Comparing an average of several scores to an individual score (whether the individual is a human or an algorithm) may give a misleading impression. In the results for any individual person, there are typically several 100% scores and a few scores in the 55-75% range. The average mapping problem has seven terms. It is not possible to have exactly one term mapped incorrectly; if there are any incorrect mappings, then there must be two or more incorrect mappings. This follows from the nature of bijections. Therefore a score of $5/7 = 71.4\%$ is not uncommon.

Table 7 looks at the results from another perspective. The column labeled *LRME wrong* gives the number of incorrect mappings made by LRME for each of the twenty problems. The five columns labeled *Number of people with N wrong* show, for various values of N, how may of the 22 people made N incorrect mappings. For the average mapping problem, 15 out of 22 participants had a perfect score ($N = 0$); of the remaining 7 participants, 5 made only two mistakes ($N = 2$). Table 7 shows more clearly than Table 6 that LRME's performance is not significantly different from (individual) human performance. (For yet another perspective, see Section 9.1).

Mapping	**LRME wrong**	**Number of people with N wrong** $N = 0$	$N = 1$	$N = 2$	$N = 3$	$N \geq 4$	m
A1	0	16	0	4	2	0	7
A2	0	14	0	5	0	3	8
A3	0	9	0	9	2	2	8
A4	0	9	0	9	0	4	8
A5	2	10	0	7	2	3	7
A6	0	20	0	2	0	0	7
A7	2	8	0	6	6	2	7
A8	0	13	0	8	0	1	8
A9	4	11	0	7	2	2	9
A10	0	13	0	9	0	0	5
M1	2	17	0	5	0	0	7
M2	0	19	0	3	0	0	7
M3	0	14	0	8	0	0	6
M4	0	22	0	0	0	0	7
M5	0	9	0	11	0	2	6
M6	0	15	0	4	3	0	7
M7	0	21	0	1	0	0	7
M8	2	18	0	2	1	1	5
M9	0	19	0	3	0	0	8
M10	0	13	0	3	3	3	6
Average	1	15	0	5	1	1	7

Table 7: Another way of viewing LRME versus human performance.

In Table 8, we examine the sensitivity of LRME to the parameter settings. The first row shows the accuracy of the baseline configuration, as in Table 6. The next eight rows show the impact of varying k, the dimensionality of the truncated singular value decomposition, from 50 to 400. The eight rows after that show the effect of varying t, the column factor, from 5 to 40. The number of columns in the matrix (n_c) is given by the number of rows (n_r = 1,662) multiplied by t. The second last row shows the effect of eliminating the singular value decomposition from LRME. This is equivalent to setting k to 1,662, the number of rows in the matrix. The final row gives the result when PPMIC (Bullinaria & Levy, 2007) is replaced with log entropy (Turney, 2006). LRME is not sensitive to any of these manipulations: None of the variations in Table 8 perform significantly differently from the baseline configuration (paired t-test, 95% confidence level). (This does not necessarily mean that the manipulations have no effect; rather, it suggests that a larger sample of problems would be needed to show a significant effect.)

Experiment	k	t	n_c	**Accuracy**
baseline configuration	300	20	33,240	91.5
varying k	50	20	33,240	89.3
	100	20	33,240	92.8
	150	20	33,240	91.3
	200	20	33,240	92.6
	250	20	33,240	90.6
	300	20	33,240	91.5
	350	20	33,240	90.6
	400	20	33,240	90.6
varying t	300	5	8,310	86.9
	300	10	16,620	94.0
	300	15	24,930	94.0
	300	20	33,240	91.5
	300	25	41,550	90.1
	300	30	49,860	90.6
	300	35	58,170	89.5
	300	40	66,480	91.7
dropping SVD	1662	20	33,240	89.7
log entropy	300	20	33,240	83.9

Table 8: Exploring the sensitivity of LRME to various parameter settings and modifications.

8. Attribute Mapping Approaches

In this section, we explore a variety of attribute mapping approaches for the twenty mapping problems. All of these approaches seek the mapping M_{a} that maximizes the sum of the attributional similarities.

$$M_{\mathrm{a}} = \underset{M \in P(A,B)}{\arg\max} \sum_{i=1}^{m} \mathrm{sim_a}(a_i, M(a_i)) \tag{36}$$

We search for M_{a} by exhaustively evaluating all of the possibilities. Ties are broken randomly. We use a variety of different algorithms to calculate $\mathrm{sim_a}$.

8.1 Algorithms

In the following experiments, we test five lexicon-based attributional similarity measures that use WordNet:[7] HSO (Hirst & St-Onge, 1998), JC (Jiang & Conrath, 1997), LC (Leacock & Chodrow, 1998), LIN (Lin, 1998), and RES (Resnik, 1995). All five are implemented in the Perl package WordNet::Similarity,[8] which builds on the WordNet::QueryData[9] package. The core idea behind them is to treat WordNet as a graph and measure the semantic distance between two terms by the length of the shortest path between them in the graph. Similarity increases as distance decreases.

7. WordNet was developed by a team at Princeton and it is available at http://wordnet.princeton.edu/.
8. Ted Pedersen's WordNet::Similarity package is at http://www.d.umn.edu/~tpederse/similarity.html.
9. Jason Rennie's WordNet::QueryData package is at http://people.csail.mit.edu/jrennie/WordNet/.

HSO works with nouns, verbs, adjectives, and adverbs, but JC, LC, LIN, and RES only work with nouns and verbs. We used WordNet::Similarity to try all possible parts of speech and all possible senses for each input word. Many adjectives, such as *true* and *valuable*, also have noun and verb senses in WordNet, so JC, LC, LIN, and RES are still able to calculate similarity for them. When the raw form of a word is not found in WordNet, WordNet::Similarity searches for morphological variations of the word. When there are multiple similarity scores, for multiple parts of speech and multiple senses, we select the highest similarity score. When there is no similarity score, because a word is not in WordNet, or because JC, LC, LIN, or RES could not find an alternative noun or verb form for an adjective or adverb, we set the score to zero.

We also evaluate two corpus-based attributional similarity measures: PMI-IR (Turney, 2001) and LSA (Landauer & Dumais, 1997). The core idea behind them is that "a word is characterized by the company it keeps" (Firth, 1957). The similarity of two terms is measured by the similarity of their statistical distributions in a corpus. We used the corpus of Section 7 along with Wumpus to implement PMI-IR (Pointwise Mutual Information with Information Retrieval). For LSA (Latent Semantic Analysis), we used the online demonstration.[10] We selected the *Matrix Comparison* option with the *General Reading up to 1st year college (300 factors)* topic space and the *term-to-term* comparison type. PMI-IR and LSA work with all parts of speech.

Our eighth similarity measure is based on the observation that our intended mappings map terms that have the same part of speech (see Appendix A). Let $\mathrm{POS}(a)$ be the part-of-speech tag assigned to the term a. We use part-of-speech tags to define a measure of attributional similarity, $\mathrm{sim}_{\mathrm{POS}}(a, b)$, as follows.

$$\mathrm{sim}_{\mathrm{POS}}(a, b) = \begin{cases} 100 & \text{if } a = b \\ 10 & \text{if } \mathrm{POS}(a) = \mathrm{POS}(b) \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

We hand-labeled the terms in the mapping problems with part-of-speech tags (Santorini, 1990). Automatic taggers assume that the words that are to be tagged are embedded in a sentence, but the terms in our mapping problems are not in sentences, so their tags are ambiguous. We used our knowledge of the intended mappings to manually disambiguate the part-of-speech tags for the terms, thus guaranteeing that corresponding terms in the intended mapping always have the same tags.

For each of the first seven attributional similarity measures above, we created seven more similarity measures by combining them with $\mathrm{sim}_{\mathrm{POS}}(a, b)$. For example, let $\mathrm{sim}_{\mathrm{HSO}}(a, b)$ be the Hirst and St-Onge (1998) similarity measure. We combine $\mathrm{sim}_{\mathrm{POS}}(a, b)$ and $\mathrm{sim}_{\mathrm{HSO}}(a, b)$ by simply adding them.

$$\mathrm{sim}_{\mathrm{HSO+POS}}(a, b) = \mathrm{sim}_{\mathrm{HSO}}(a, b) + \mathrm{sim}_{\mathrm{POS}}(a, b) \tag{38}$$

The values returned by $\mathrm{sim}_{\mathrm{POS}}(a, b)$ range from 0 to 100, whereas the values returned by $\mathrm{sim}_{\mathrm{HSO}}(a, b)$ are much smaller. We chose large values in (37) so that getting POS tags to match up has more weight than any of the other similarity measures. The manual POS tags

10. The online demonstration of LSA is the work of a team at the University of Colorado at Boulder. It is available at http://lsa.colorado.edu/.

and the high weight of $\mathrm{sim}_{\mathrm{POS}}(a, b)$ give an unfair advantage to the attributional mapping approach, but the relational mapping approach can afford to be generous.

8.2 Experiments

Table 9 presents the accuracy of the various measures of attributional similarity. The best result without POS labels is 55.9% (HSO). The best result with POS labels is 76.8% (LIN+POS). The 91.5% accuracy of LRME (see Table 6) is significantly higher than the 76.8% accuracy of LIN+POS (and thus, of course, significantly higher than everything else in Table 9; paired t-test, 95% confidence level). The average human performance of 87.6% (see Table 5) is also significantly higher than the 76.8% accuracy of LIN+POS (paired t-test, 95% confidence level). In summary, humans and LRME perform significantly better than all of the variations of attributional mapping approaches that were tested.

Algorithm	**Reference**	**Accuracy**
HSO	Hirst and St-Onge (1998)	55.9
JC	Jiang and Conrath (1997)	54.7
LC	Leacock and Chodrow (1998)	48.5
LIN	Lin (1998)	48.2
RES	Resnik (1995)	43.8
PMI-IR	Turney (2001)	54.4
LSA	Landauer and Dumais (1997)	39.6
POS (hand-labeled)	Santorini (1990)	44.8
HSO+POS	Hirst and St-Onge (1998)	71.1
JC+POS	Jiang and Conrath (1997)	73.6
LC+POS	Leacock and Chodrow (1998)	69.5
LIN+POS	Lin (1998)	76.8
RES+POS	Resnik (1995)	71.6
PMI-IR+POS	Turney (2001)	72.8
LSA+POS	Landauer and Dumais (1997)	65.8

Table 9: The accuracy of attribute mapping approaches for a wide variety of measures of attributional similarity.

9. Discussion

In this section, we examine three questions that are suggested by the preceding results. Is there a difference between the science analogy problems and the common metaphor problems? Is there an advantage to combining the relational and attributional mapping approaches? What is the advantage of the relational mapping approach over the attributional mapping approach?

9.1 Science Analogies versus Common Metaphors

Table 5 suggests that science analogies may be more difficult than common metaphors. This is supported by Table 10, which shows how the agreement of the 22 participants with our intended mapping (see Section 6) varies between the science problems and the metaphor problems. The science problems have a lower average performance and greater variation in performance. The difference between the science problems and the metaphor problems is statistically significant (paired t-test, 95% confidence level).

	Average Accuracy		
Participant	**All 20**	**10 Science**	**10 Metaphor**
1	72.6	59.9	85.4
2	88.2	85.9	90.5
3	90.0	86.3	**93.8**
4	71.8	56.4	87.1
5	**95.7**	**94.2**	**97.1**
6	83.4	83.9	82.9
7	79.6	73.6	85.7
8	**91.9**	**95.0**	88.8
9	89.7	**90.0**	89.3
10	80.7	81.4	80.0
11	**94.5**	**95.7**	**93.3**
12	90.6	87.4	**93.8**
13	**93.2**	89.6	**96.7**
14	**97.1**	**94.3**	**100.0**
15	86.6	88.5	84.8
16	80.5	80.2	80.7
17	**93.3**	**89.9**	**96.7**
18	86.5	78.9	**94.2**
19	**92.9**	**96.0**	89.8
20	90.4	84.1	**96.7**
21	82.7	74.9	90.5
22	**96.2**	**94.9**	**97.5**
Average	87.6	84.6	90.7
Standard deviation	7.2	10.8	5.8

Table 10: A comparison of the difficulty of the science problems versus the metaphor problems for the 22 participants. The numbers in bold font are the scores that are above the scores of LRME.

The average science problem has more terms (7.4) than the average metaphor problem (6.6), which might contribute to the difficulty of the science problems. However, Table 11 shows that there is no clear relation between the number of terms in a problem (m in Table 5) and the level of agreement. We believe that people find the metaphor problems easier than the science problems because these common metaphors are entrenched in our language, whereas the science analogies are more peripheral.

Table 12 shows that the 16 algorithms studied here perform slightly worse on the science problems than on the metaphor problems, but the difference is not statistically significant (paired t-test, 95% confidence level). We hypothesize that the attributional mapping approaches are not performing well enough to be sensitive to subtle differences between science analogies and common metaphors.

Incidentally, these tables give us another view of the performance of LRME in comparison to human performance. The first row in Table 12 shows the performance of LRME on

Num terms	Agreement
5	86.4
6	81.3
7	91.1
8	86.5
9	84.3

Table 11: The average agreement among the 22 participants as a function of the number of terms in the problems.

	Average Accuracy		
Algorithm	**All 20**	**10 Science**	**10 Metaphor**
LRME	91.5	89.8	93.1
HSO	55.9	57.4	54.3
JC	54.7	57.4	52.1
LC	48.5	49.6	47.5
LIN	48.2	46.7	49.7
RES	43.8	39.0	48.6
PMI-IR	54.4	49.5	59.2
LSA	39.6	37.3	41.9
POS	44.8	42.1	47.4
HSO+POS	71.1	66.9	75.2
JC+POS	73.6	78.1	69.2
LC+POS	69.5	70.8	68.2
LIN+POS	76.8	68.8	84.8
RES+POS	71.6	70.3	72.9
PMI-IR+POS	72.8	65.7	79.9
LSA+POS	65.8	69.1	62.4
Average	61.4	59.9	62.9
Standard deviation	14.7	15.0	15.3

Table 12: A comparison of the difficulty of the science problems versus the metaphor problems for the 16 algorithms.

the science and metaphor problems. In Table 10, we have marked in bold font the cases where human scores are greater than LRME's scores. For all 20 problems, there are 8 such cases; for the 10 science problems, there are 8 such cases; for the 10 metaphor problems, there are 10 such cases. This is further evidence that LRME's performance is not significantly different from human performance. LRME is near the middle of the range of performance of the 22 human participants.

9.2 Hybrid Relational-Attributional Approaches

Recall the definitions of $\mathrm{score_r}(M)$ and $\mathrm{score_a}(M)$ given in Section 3.

$$\text{score}_\text{r}(M) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \text{sim}_\text{r}(a_i\!:\!a_j, M(a_i)\!:\!M(a_j)) \tag{39}$$

$$\text{score}_\text{a}(M) = \sum_{i=1}^{m} \text{sim}_\text{a}(a_i, M(a_i)) \tag{40}$$

We can combine the scores by simply adding them or multiplying them, but $\text{score}_\text{r}(M)$ and $\text{score}_\text{a}(M)$ may be quite different in the scales and distributions of their values; therefore we first normalize them to probabilities.

$$\text{prob}_\text{r}(M) = \frac{\text{score}_\text{r}(M)}{\sum_{M_i \in P(A,B)} \text{score}_\text{r}(M_i)} \tag{41}$$

$$\text{prob}_\text{a}(M) = \frac{\text{score}_\text{a}(M)}{\sum_{M_i \in P(A,B)} \text{score}_\text{a}(M_i)} \tag{42}$$

For these probability estimates, we assume that $\text{score}_\text{r}(M) \geq 0$ and $\text{score}_\text{a}(M) \geq 0$. If necessary, a constant value may be added to the scores, to ensure that they are not negative. Now we can combine the scores by adding or multiplying the probabilities.

$$M_{\text{r+a}} = \underset{M \in P(A,B)}{\arg\max} \big(\text{prob}_\text{r}(M) + \text{prob}_\text{a}(M)\big) \tag{43}$$

$$M_{\text{r}\times\text{a}} = \underset{M \in P(A,B)}{\arg\max} \big(\text{prob}_\text{r}(M) \times \text{prob}_\text{a}(M)\big) \tag{44}$$

Table 13 shows the accuracy when LRME is combined with LIN+POS (the best attributional mapping algorithm in Table 9, with an accuracy of 76.8%) or with HSO (the best attributional mapping algorithm that does not use the manual POS tags, with an accuracy of 55.9%). We try both adding and multiplying probabilities. On its own, LRME has an accuracy of 91.5%. Combining LRME with LIN+POS increases the accuracy to 94.0%, but this improvement is not statistically significant (paired t-test, 95% confidence level). Combining LRME with HSO results in a decrease in accuracy. The decrease is not significant when the probabilities are multiplied (85.4%), but it is significant when the probabilities are added (78.5%).

In summary, the experiments show no significant advantage to combining LRME with attributional mapping. However, it is possible that a larger sample of problems would show a significant advantage. Also, the combination methods we explored (addition and multiplication of probabilities) are elementary. A more sophisticated approach, such as a weighted combination, may perform better.

9.3 Coherent Relations

We hypothesize that LRME benefits from a kind of coherence among the relations. On the other hand, attributional mapping approaches do not involve this kind of coherence.

Components			
Relational	**Attributional**	**Combination**	**Accuracy**
LRME	LIN+POS	add probabilities	94.0
LRME	LIN+POS	multiply probabilities	94.0
LRME	HSO	add probabilities	78.5
LRME	HSO	multiply probabilities	85.4

Table 13: The performance of four different hybrids of relational and attributional mapping approaches.

Suppose we swap two of the terms in a mapping. Let M be the original mapping and let M' be the new mapping, where $M'(a_1) = M(a_2)$, $M'(a_2) = M(a_1)$, and $M'(a_i) = M(a_i)$ for $i > 2$. With attributional similarity, the impact of this swap on the score of the mapping is limited. Part of the score is not affected.

$$\text{score}_\text{a}(M) = \text{sim}_\text{a}(a_1, M(a_1)) + \text{sim}_\text{a}(a_2, M(a_2)) + \sum_{i=3}^{m} \text{sim}_\text{a}(a_i, M(a_i)) \tag{45}$$

$$\text{score}_\text{a}(M') = \text{sim}_\text{a}(a_1, M(a_2)) + \text{sim}_\text{a}(a_2, M(a_1)) + \sum_{i=3}^{m} \text{sim}_\text{a}(a_i, M(a_i)) \tag{46}$$

On the other hand, with relational similarity, the impact of a swap is not limited in this way. A change to any part of the mapping affects the whole score. There is a kind of global coherence to relational similarity that is lacking in attributional similarity.

Testing the hypothesis that LRME benefits from coherence is somewhat complicated, because we need to design the experiment so that the coherence effect is isolated from any other effects. To do this, we move some of the terms outside of the accuracy calculation.

Let $M_* : A \rightarrow B$ be one of our twenty mapping problems, where M_* is our intended mapping and $m = |A| = |B|$. Let A' be a randomly selected subset of A of size m'. Let B' be $M_*(A')$, the subset of B to which M_* maps A'.

$$A' \subset A \tag{47}$$
$$B' \subset B \tag{48}$$
$$B' = M_*(A') \tag{49}$$
$$m' = |A'| = |B'| \tag{50}$$
$$m' < m \tag{51}$$

There are two ways that we might use LRME to generate a mapping $M' : A' \rightarrow B'$ for this new reduced mapping problem, *internal coherence* and *total coherence*.

1. **Internal coherence:** We can select M' based on $\langle A', B' \rangle$ alone.

$$A' = \{a_1, ..., a_{m'}\} \tag{52}$$
$$B' = \{b_1, ..., b_{m'}\} \tag{53}$$
$$M' = \operatorname*{arg\,max}_{M \in P(A',B')} \sum_{i=1}^{m'} \sum_{j=i+1}^{m'} \mathrm{sim_r}(a_i\!:\!a_j, M(a_i)\!:\!M(a_j)) \tag{54}$$

In this case, M' is chosen based only on the relations that are internal to $\langle A', B' \rangle$.

2. **Total coherence:** We can select M' based on $\langle A, B \rangle$ and the knowledge that M' must satisfy the constraint that $M'(A') = B'$. (This knowledge is also embedded in internal coherence.)

$$A = \{a_1, ..., a_m\} \tag{55}$$
$$B = \{b_1, ..., b_m\} \tag{56}$$
$$P'(A, B) = \big\{M |\ M \in P(A, B) \text{ and } M(A') = B'\big\} \tag{57}$$
$$M' = \operatorname*{arg\,max}_{M \in P'(A,B)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mathrm{sim_r}(a_i\!:\!a_j, M(a_i)\!:\!M(a_j)) \tag{58}$$

In this case, M' is chosen using both the relations that are internal to $\langle A', B' \rangle$ and other relations in $\langle A, B \rangle$ that are external to $\langle A', B' \rangle$.

Suppose that we calculate the accuracy of these two methods based only on the subproblem $\langle A', B' \rangle$. At first it might seem that there is no advantage to total coherence, because it must explore a larger space of possible mappings than internal coherence (since $|P'(A, B)|$ is larger than $|P(A', B')|$), but the additional terms that it explores are not involved in calculating the accuracy. However, we hypothesize that total coherence will have a higher accuracy than internal coherence, because the additional external relations help to select the correct mapping.

To test this hypothesis, we set m' to 3 and we randomly generated ten new reduced mapping problems for each of the twenty problems (i.e., a total of 200 new problems of size 3). The average accuracy of internal coherence was 93.3%, whereas the average accuracy of total coherence was 97.3%. The difference is statistically significant (paired t-test, 95% confidence level).

On the other hand, the attributional mapping approaches cannot benefit from total coherence, because there is no connection between the attributes that are in $\langle A', B' \rangle$ and the attributes that are outside. We can decompose $\mathrm{score_a}(M)$ into two independent parts.

$$
\begin{aligned}
A'' &= A \setminus A' && (59)\\
A &= A' \cup A'' && (60)\\
P'(A,B) &= \left\{ M |\; M \in P(A,B) \text{ and } M(A') = B' \right\} && (61)\\
M' &= \underset{M \in P'(A,B)}{\arg\max} \sum_{a_i \in A} \mathrm{sim_a}(a_i, M(a_i)) && (62)\\
&= \underset{M \in P'(A,B)}{\arg\max} \left(\sum_{a_i \in A'} \mathrm{sim_a}(a_i, M(a_i)) + \sum_{a_i \in A''} \mathrm{sim_a}(a_i, M(a_i)) \right) && (63)
\end{aligned}
$$

These two parts can be optimized independently. Thus the terms that are external to $\langle A', B' \rangle$ have no influence on the part of M' that covers $\langle A', B' \rangle$.

Relational mapping cannot be decomposed into independent parts in this way, because the relations connect the parts. This gives relational mapping approaches an inherent advantage over attributional mapping approaches.

To confirm this analysis, we compared internal and total coherence using LIN+POS on the same 200 new problems of size 3. The average accuracy of internal coherence was 88.0%, whereas the average accuracy of total coherence was 87.0%. The difference is not statistically significant (paired t-test, 95% confidence level). (The only reason that there is any difference is that, when two mappings have the same score, we break the ties randomly. This causes random variation in the accuracy.)

The benefit from coherence suggests that we can make analogy mapping problems easier for LRME by adding more terms. The difficulty is that the new terms cannot be randomly chosen; they must fit with the logic of the analogy and not overlap with the existing terms.

Of course, this is not the only important difference between the relational and attributional mapping approaches. We believe that the most important difference is that relations are more reliable and more general than attributes, when using past experiences to make predictions about the future (Hofstadter, 2001; Gentner, 2003). Unfortunately, this hypothesis is more difficult to evaluate experimentally than our hypothesis about coherence.

10. Related Work

French (2002) gives a good survey of computational approaches to analogy-making, from the perspective of cognitive science (where the emphasis is on how well computational systems model human performance, rather than how well the systems perform). We will sample a few systems from his survey and add a few more that were not mentioned.

French (2002) categorizes analogy-making systems as *symbolic*, *connectionist*, or *symbolic-connectionist hybrids*. Gärdenfors (2004) proposes another category of representational systems for AI and cognitive science, which he calls *conceptual spaces*. These spatial or geometric systems are common in information retrieval and machine learning (Widdows, 2004; van Rijsbergen, 2004). An influential example is Latent Semantic Analysis (Landauer & Dumais, 1997). The first spatial approaches to analogy-making began to appear around the same time as French's (2002) survey. LRME takes a spatial approach to analogy-making.

10.1 Symbolic Approaches

Computational approaches to analogy-making date back to ANALOGY (Evans, 1964) and Argus (Reitman, 1965). Both of these systems were designed to solve proportional analogies (analogies in which $|A| = |B| = 2$; see Section 4). ANALOGY could solve proportional analogies with simple geometric figures and Argus could solve simple word analogies. These systems used hand-coded rules and were only able to solve the limited range of problems that their designers had anticipated and coded in the rules.

French (2002) cites Structure Mapping Theory (SMT) (Gentner, 1983) and the Structure Mapping Engine (SME) (Falkenhainer et al., 1989) as the prime examples of symbolic approaches:

> SMT is unquestionably the most influential work to date on the modeling of analogy-making and has been applied in a wide range of contexts ranging from child development to folk physics. SMT explicitly shifts the emphasis in analogy-making to the structural similarity between the source and target domains. Two major principles underlie SMT:
>
> - the relation-matching principle: good analogies are determined by mappings of relations and not attributes (originally only identical predicates were mapped) and
> - the systematicity principle: mappings of coherent systems of relations are preferred over mappings of individual relations.
>
> This structural approach was intended to produce a domain-independent mapping process.

LRME follows both of these principles. LRME uses only relational similarity; no attributional similarity is involved (see Section 7.1). Coherent systems of relations are preferred over mappings of individual relations (see Section 9.3). However, the spatial (statistical, corpus-based) approach of LRME is quite different from the symbolic (logical, hand-coded) approach of SME.

Martin (1992) uses a symbolic approach to handle conventional metaphors. Gentner, Bowdle, Wolff, and Boronat (2001) argue that novel metaphors are processed as analogies, but conventional metaphors are recalled from memory without special processing. However, the line between conventional and novel metaphor can be unclear.

Dolan (1995) describes an algorithm that can extract conventional metaphors from a dictionary. A semantic parser is used to extract semantic relations from the Longman Dictionary of Contemporary English (LDOCE). A symbolic algorithm finds metaphorical relations between words, using the extracted relations.

Veale (2003, 2004) has developed a symbolic approach to analogy-making, using WordNet as a lexical resource. Using a spreading activation algorithm, he achieved a score of 43.0% on a set of 374 multiple-choice lexical proportional analogy questions from the SAT college entrance test (Veale, 2004).

Lepage (1998) has demonstrated that a symbolic approach to proportional analogies can be used for morphology processing. Lepage and Denoual (2005) apply a similar approach to machine translation.

10.2 Connectionist Approaches

Connectionist approaches to analogy-making include ACME (Holyoak & Thagard, 1989) and LISA (Hummel & Holyoak, 1997). Like symbolic approaches, these systems use hand-coded knowledge representations, but the search for mappings takes a connectionist approach, in which there are nodes with weights that are incrementally updated over time, until the system reaches a stable state.

10.3 Symbolic-Connectionist Hybrid Approaches

The third family examined by French (2002) is hybrid approaches, containing elements of both the symbolic and connectionist approaches. Examples include Copycat (Mitchell, 1993) and Tabletop (French, 1995). Much of the work in the Fluid Analogies Research Group (FARG) concerns symbolic-connectionist hybrids (Hofstadter & FARG, 1995).

10.4 Spatial Approaches

Marx, Dagan, Buhmann, and Shamir (2002) present the *coupled clustering* algorithm, which uses a feature vector representation to find analogies in collections of text. For example, given documents on Buddhism and Christianity, it finds related terms, such as {*school, Mahayana, Zen*} for Buddhism and {*tradition, Catholic, Protestant*} for Christianity.

Mason (2004) describes the CorMet system for extracting conventional metaphors from text. CorMet is based on clustering feature vectors that represent the selectional preferences of verbs. Given keywords for the source domain *laboratory* and the target domain *finance*, it is able to discover mappings such as *liquid* → *income* and *container* → *institution*.

Turney, Littman, Bigham, and Shnayder (2003) present a system for solving lexical proportional analogy questions from the SAT college entrance test, which combines thirteen different modules. Twelve of the modules use either attributional similarity or a symbolic approach to relational similarity, but one module uses a spatial (feature vector) approach to measuring relational similarity. This module worked much better than any of the other modules; therefore, it was studied in more detail by Turney and Littman (2005). The relation between a pair of words is represented by a vector, in which the elements are pattern frequencies. This is similar to LRME, but one important difference is that Turney and Littman (2005) used a fixed, hand-coded set of 128 patterns, whereas LRME automatically generates a variable number of patterns from the given corpus (33,240 patterns in our experiments here).

Turney (2005) introduced Latent Relational Analysis (LRA), which was examined more thoroughly by Turney (2006). LRA achieves human-level performance on a set of 374 multiple-choice proportional analogy questions from the SAT college entrance exam. LRME uses a simplified form of LRA. A similar simplification of LRA is used by Turney (2008), in a system for processing analogies, synonyms, antonyms, and associations. The contribution of LRME is to go beyond proportional analogies, to larger systems of analogical mappings.

10.5 General Theories of Analogy and Metaphor

Many theories of analogy-making and metaphor either do not involve computation or they suggest general principles and concepts that are not specific to any particular computational approach. The design of LRME has been influenced by several theories of this type (Gentner, 1983; Hofstadter & FARG, 1995; Holyoak & Thagard, 1995; Hofstadter, 2001; Gentner, 2003).

Lakoff and Johnson (1980) provide extensive evidence that metaphor is ubiquitous in language and thought. We believe that a system for analogy-making should be able to handle metaphorical language, which is why ten of our analogy problems are derived from Lakoff and Johnson (1980). We agree with their claim that a metaphor does not merely involve a superficial relation between a couple of words; rather, it involves a systematic set of mappings between two domains. Thus our analogy problems involve larger sets of words, beyond proportional analogies.

Holyoak and Thagard (1995) argue that analogy-making is central in our daily thought, and especially in finding creative solutions to new problems. Our ten scientific analogies were derived from their examples of analogy-making in scientific creativity.

11. Limitations and Future Work

In Section 4, we mentioned ten applications for LRA, and in Section 5 we claimed that the results of the experiments in Section 9.3 suggest that LRME may perform better than LRA on all ten of these applications, due to its ability to handle bijective analogies when $m > 2$. Our focus in future work will be testing this hypothesis. In particular, the task of semantic role labeling, discussed in Section 1, seems to be a good candidate application for LRME.

The input to LRME is simpler than the input to SME (compare Figures 1 and 2 in Section 1 with Table 1), but there is still some human effort involved in creating the input. LRME is not immune to the criticism of Chalmers, French, and Hofstadter (1992), that the human who generates the input is doing more work than the computer that makes the mappings, although it is not a trivial matter to find the right mapping out of 5,040 (7!) choices.

In future work, we would like to relax the requirement that $\langle A, B \rangle$ must be a bijection (see Section 3), by adding irrelevant words (distractors) and synonyms. The mapping algorithm will be forced to decide what terms to include in the mapping and what terms to leave out.

We would also like to develop an algorithm that can take a proportional analogy ($m = 2$) as input (e.g., sun:planet::nucleus:electron) and automatically expand it to a larger analogy ($m > 2$, e.g., Table 2). That is, it would automatically search the corpus for new terms to add to the analogy.

The next step would be to give the computer only the topic of the source domain (e.g., solar system) and the topic of the target domain (e.g., atomic structure), and let it work out the rest on its own. This might be possible by combining ideas from LRME with ideas from coupled clustering (Marx et al., 2002) and CorMet (Mason, 2004).

It seems that analogy-making is triggered in people when we encounter a problem (Holyoak & Thagard, 1995). The problem defines the target for us, and we immediately start searching for a source. Analogical mapping enables us to transfer our knowledge of the source to the target, hopefully leading to a solution to the problem. This suggests that the input to the ideal analogical mapping algorithm would be simply a statement that there is a problem (e.g., What is the structure of the atom?). Ultimately, the computer might find the problems on its own as well. The only input would be a large corpus.

The algorithms we have considered here all perform exhaustive search of the set of possible mappings $P(A, B)$. This is acceptable when the sets are small, as they are here, but it will be problematic for larger problems. In future work, it will be necessary to use heuristic search algorithms instead of exhaustive search.

It takes almost 18 hours for LRME to process the twenty mapping problems (Section 7). With better hardware and some changes to the software, this time could be significantly reduced. For even greater speed, the algorithm could run continuously, building a large database of vector representations of term pairs, so that it is ready to create mappings as soon as a user requests them. This is similar to the vision of Banko and Etzioni (2007).

LRME, like LRA and LSA (Landauer & Dumais, 1997), uses a truncated singular value decomposition (SVD) to smooth the matrix. Many other algorithms have been proposed for smoothing matrices. In our past work with LRA (Turney, 2006), we experimented with Nonnegative Matrix Factorization (NMF) (Lee & Seung, 1999), Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999), Iterative Scaling (IS) (Ando, 2000), and Kernel Principal Components Analysis (KPCA) (Scholkopf, Smola, & Muller, 1997). We had some interesting results with small matrices (around 1000 $\times$ 2000), but none of the algorithms seemed substantially better than truncated SVD, and none of them scaled up to the matrix sizes that we have here (1,662 $\times$ 33,240). However, we believe that SVD is not unique, and future work is likely to discover a smoothing algorithm that is more efficient and effective than SVD. The results in Section 7.2 do not show a significant benefit from SVD. Table 8 hints that PPMIC (Bullinaria & Levy, 2007) is more important than SVD.

LRME extracts knowledge from many fragments of text. In Section 7.1, we noted that we found an average of 1,180 phrases per pair. The information from these 1,180 phrases is combined in a vector, to represent the semantic relation for a pair. This is quite different from relation extraction in (for example) the Automatic Content Extraction (ACE) Evaluation.[11] The task in ACE is to identify and label a semantic relation in a single sentence. Semantic role labeling also involves labeling a single sentence (Gildea & Jurafsky, 2002).

The contrast between LRME and ACE is analogous to the distinction in cognitive psychology between semantic and episodic memory. Episodic memory is memory of a specific event in one's personal past, whereas semantic memory is memory of basic facts and concepts, unrelated to any specific event in the past. LRME extracts relational information that is independent of any specific sentence, like semantic memory. ACE is concerned with extracting the relation in a specific sentence, like episodic memory. In cognition, episodic memory and semantic memory work together synergistically. When we experience an event, we use our semantic memory to interpret the event and form a new episodic memory, but semantic memory is itself constructed from our past experiences, our accumulated episodic memories. This suggests that there should be a synergy from combining LRME-like semantic information extraction algorithms with ACE-like episodic information extraction algorithms.

11. ACE is an annual event that began in 1999. Relation Detection and Characterization (RDC) was introduced to ACE in 2001. For more information, see http://www.nist.gov/speech/tests/ace/.

12. Conclusion

Analogy is the core of cognition. We understand the present by analogy to the past. We predict the future by analogy to the past and the present. We solve problems by searching for analogous situations (Holyoak & Thagard, 1995). Our daily language is saturated with metaphor (Lakoff & Johnson, 1980), and metaphor is based on analogy (Gentner et al., 2001). To understand human language, to solve human problems, to work with humans, computers must be able to make analogical mappings.

Our best theory of analogy-making is Structure Mapping Theory (Gentner, 1983), but the Structure Mapping Engine (Falkenhainer et al., 1989) puts too much of the burden of analogy-making on its human users (Chalmers et al., 1992). LRME is an attempt to shift some of that burden onto the computer, while remaining consistent with the general principles of SMT.

We have shown that LRME is able to solve bijective analogical mapping problems with human-level performance. Attributional mapping algorithms (at least, those we have tried so far) are not able to reach this level. This supports SMT, which claims that relations are more important than attributes when making analogical mappings.

There is still much research to be done. LRME takes some of the load off the human user, but formulating the input to LRME is not easy. This paper is an incremental step towards a future in which computers can make surprising and useful analogies with minimal human assistance.

Acknowledgments

Thanks to my colleagues at the Institute for Information Technology for participating in the experiment in Section 6. Thanks to Charles Clarke and Egidio Terra for their corpus. Thanks to Stefan Büttcher for making Wumpus available and giving me advice on its use. Thanks to Doug Rohde for making SVDLIBC available. Thanks to the WordNet team at Princeton University for WordNet, Ted Pedersen for the WordNet::Similarity Perl package, and Jason Rennie for the WordNet::QueryData Perl package. Thanks to the LSA team at the University of Colorado at Boulder for the use of their online demonstration of LSA. Thanks to Deniz Yuret, André Vellino, Dedre Gentner, Vivi Nastase, Yves Lepage, Diarmuid Ó Séaghdha, Roxana Girju, Chris Drummond, Howard Johnson, Stan Szpakowicz, and the anonymous reviewers of *JAIR* for their helpful comments and suggestions.

Appendix A. Details of the Mapping Problems

In this appendix, we provide detailed information about the twenty mapping problems. Figure 3 shows the instructions that were given to the participants in the experiment in Section 6. These instructions were displayed in their web browsers. Tables 14, 15, 16, and 17 show the twenty mapping problems. The first column gives the problem number (e.g., A1) and a mnemonic that summarizes the mapping (e.g., solar system → atom). The second column gives the source terms and the third column gives the target terms.

The mappings shown in these tables are our intended mappings. The fourth column shows the percentage of participants who agreed with our intended mappings. For example,

Systematic Analogies and Metaphors

Instructions

You will be presented with twenty analogical mapping problems, ten based on scientific analogies and ten based on common metaphors. A typical problem will look like this:

horse	→	? ∇
legs	→	? ∇
hay	→	? ∇
brain	→	? ∇
dung	→	? ∇

You may click on the drop-down menus above, to see what options are available.

Your task is to construct an analogical mapping; that is, a one-to-one mapping between the items on the left and the items on the right. For example:

horse	→	car ∇
legs	→	wheels ∇
hay	→	gasoline ∇
brain	→	driver ∇
dung	→	exhaust ∇

This mapping expresses an analogy between a horse and a car. The horse's legs are like the car's wheels. The horse eats hay and the car consumes gasoline. The horse's brain controls the movement of the horse like the car's driver controls the movement of the car. The horse generates dung as a waste product like the car generates exhaust as a waste product.

You should have no duplicate items in your answers on the right-hand side. If there are any duplicates or missing items (question marks), you will get an error message when you submit your answer.

You are welcome to use a dictionary as you work on the problems, if you would find it helpful.

If you find the above instructions unclear, then please do not continue with this exercise. Your answers to the twenty problems will be used as a standard for evaluating the output of a computer algorithm; therefore, you should only proceed if you are confident that you understand this task.

Figure 3: The instructions for the participants in the experiment in Section 6.

Mapping	**Source**	**→ Target**	**Agreement**	**POS**
	solar system	→ atom	86.4	NN
A1	sun	→ nucleus	100.0	NN
	planet	→ electron	95.5	NN
solar system	mass	→ charge	86.4	NN
→ atom	attracts	→ attracts	90.9	VBZ
	revolves	→ revolves	95.5	VBZ
	gravity	→ electromagnetism	81.8	NN
	Average agreement:		90.9	
	water	→ heat	86.4	NN
A2	flows	→ transfers	95.5	VBZ
	pressure	→ temperature	86.4	NN
water flow	water tower	→ burner	72.7	NN
→ heat transfer	bucket	→ kettle	72.7	NN
	filling	→ heating	95.5	VBG
	emptying	→ cooling	95.5	VBG
	hydrodynamics	→ thermodynamics	90.9	NN
	Average agreement:		86.9	
	waves	→ sounds	86.4	NNS
A3	shore	→ wall	77.3	NN
	reflects	→ echoes	95.5	VBZ
waves	water	→ air	95.5	NN
→ sounds	breakwater	→ insulation	81.8	NN
	rough	→ loud	63.6	JJ
	calm	→ quiet	100.0	JJ
	crashing	→ vibrating	54.5	VBG
	Average agreement:		81.8	
	combustion	→ respiration	72.7	NN
A4	fire	→ animal	95.5	NN
	fuel	→ food	90.9	NN
combustion	burning	→ breathing	72.7	VBG
→ respiration	hot	→ living	59.1	JJ
	intense	→ vigorous	77.3	JJ
	oxygen	→ oxygen	77.3	NN
	carbon dioxide	→ carbon dioxide	86.4	NN
	Average agreement:		79.0	
	sound	→ light	86.4	NN
A5	low	→ red	50.0	JJ
	high	→ violet	54.5	JJ
sound	echoes	→ reflects	100.0	VBZ
→ light	loud	→ bright	90.9	JJ
	quiet	→ dim	77.3	JJ
	horn	→ lens	95.5	NN
	Average agreement:		79.2	

Table 14: Science analogy problems A1 to A5, derived from Chapter 8 of Holyoak and Thagard (1995).

Mapping	Source	→ Target	Agreement	POS
A6	projectile	→ planet	100.0	NN
	trajectory	→ orbit	100.0	NN
projectile	earth	→ sun	100.0	NN
→ planet	parabolic	→ elliptical	100.0	JJ
	air	→ space	100.0	NN
	gravity	→ gravity	90.9	NN
	attracts	→ attracts	90.9	VBZ
	Average agreement:		97.4	
A7	breeds	→ species	100.0	NNS
	selection	→ competition	59.1	NN
artificial selection	conformance	→ adaptation	59.1	NN
→ natural selection	artificial	→ natural	77.3	JJ
	popularity	→ fitness	54.5	NN
	breeding	→ mating	95.5	VBG
	domesticated	→ wild	77.3	JJ
	Average agreement:		74.7	
A8	balls	→ molecules	90.9	NNS
	billiards	→ gas	72.7	NN
billiard balls	speed	→ temperature	81.8	NN
→ gas molecules	table	→ container	95.5	NN
	bouncing	→ pressing	77.3	VBG
	moving	→ moving	86.4	VBG
	slow	→ cold	100.0	JJ
	fast	→ hot	100.0	JJ
	Average agreement:		88.1	
A9	computer	→ mind	90.9	NN
	processing	→ thinking	95.5	VBG
computer	erasing	→ forgetting	100.0	VBG
→ mind	write	→ memorize	72.7	VB
	read	→ remember	54.5	VB
	memory	→ memory	81.8	NN
	outputs	→ muscles	72.7	NNS
	inputs	→ senses	90.9	NNS
	bug	→ mistake	100.0	NN
	Average agreement:		84.3	
A10	slot machines	→ bacteria	68.2	NNS
	reels	→ genes	72.7	NNS
slot machine	spinning	→ mutating	86.4	VBG
→ bacterial mutation	winning	→ reproducing	90.9	VBG
	losing	→ dying	100.0	VBG
	Average agreement:		83.6	

Table 15: Science analogy problems A6 to A10, derived from Chapter 8 of Holyoak and Thagard (1995).

Mapping	Source	→ Target	Agreement	POS
	war	→ argument	90.9	NN
M1	soldier	→ debater	100.0	NN
	destroy	→ refute	90.9	VB
war	fighting	→ arguing	95.5	VBG
→ argument	defeat	→ acceptance	90.9	NN
	attacks	→ criticizes	95.5	VBZ
	weapon	→ logic	90.9	NN
	Average agreement:		93.5	
	buyer	→ believer	100.0	NN
M2	merchandise	→ belief	90.9	NN
	buying	→ accepting	95.5	VBG
buying an item	selling	→ advocating	100.0	VBG
→ accepting a belief	returning	→ rejecting	95.5	VBG
	valuable	→ true	95.5	JJ
	worthless	→ false	95.5	JJ
	Average agreement:		96.1	
	foundations	→ reasons	72.7	NNS
M3	buildings	→ theories	77.3	NNS
	supporting	→ confirming	95.5	VBG
grounds for a building	solid	→ rational	90.9	JJ
→ reasons for a theory	weak	→ dubious	95.5	JJ
	crack	→ flaw	95.5	NN
	Average agreement:		87.9	
	obstructions	→ difficulties	100.0	NNS
M4	destination	→ goal	100.0	NN
	route	→ plan	100.0	NN
impediments to travel	traveller	→ person	100.0	NN
→ difficulties	travelling	→ problem solving	100.0	VBG
	companion	→ partner	100.0	NN
	arriving	→ succeeding	100.0	VBG
	Average agreement:		100.0	
	money	→ time	95.5	NN
M5	allocate	→ invest	86.4	VB
	budget	→ schedule	86.4	NN
money	effective	→ efficient	86.4	JJ
→ time	cheap	→ quick	50.0	JJ
	expensive	→ slow	59.1	JJ
	Average agreement:		77.3	

Table 16: Common metaphor problems M1 to M5, derived from Lakoff and Johnson (1980).

Mapping	Source	→ Target	Agreement	POS
M6	seeds	→ ideas	90.9	NNS
seeds → ideas	planted	→ inspired	95.5	VBD
	fruitful	→ productive	81.8	JJ
	fruit	→ product	95.5	NN
	grow	→ develop	81.8	VB
	wither	→ fail	100.0	VB
	blossom	→ succeed	77.3	VB
	Average agreement:		89.0	
M7	machine	→ mind	95.5	NN
machine → mind	working	→ thinking	100.0	VBG
	turned on	→ awake	100.0	JJ
	turned off	→ asleep	100.0	JJ
	broken	→ confused	100.0	JJ
	power	→ intelligence	95.5	NN
	repair	→ therapy	100.0	NN
	Average agreement:		98.7	
M8	object	→ idea	90.9	NN
object → idea	hold	→ understand	81.8	VB
	weigh	→ analyze	81.8	VB
	heavy	→ important	95.5	JJ
	light	→ trivial	95.5	JJ
	Average agreement:		89.1	
M9	follow	→ understand	100.0	VB
following → understanding	leader	→ speaker	100.0	NN
	path	→ argument	100.0	NN
	follower	→ listener	100.0	NN
	lost	→ misunderstood	86.4	JJ
	wanders	→ digresses	90.9	VBZ
	twisted	→ complicated	95.5	JJ
	straight	→ simple	100.0	JJ
	Average agreement:		96.6	
M10	seeing	→ understanding	68.2	VBG
seeing → understanding	light	→ knowledge	77.3	NN
	illuminating	→ explaining	86.4	VBG
	darkness	→ confusion	86.4	NN
	view	→ interpretation	68.2	NN
	hidden	→ secret	86.4	JJ
	Average agreement:		78.8	

Table 17: Common metaphor problems M6 to M10, derived from Lakoff and Johnson (1980).

in problem A1, 81.8% of the participants (18 out of 22) mapped gravity to electromagnetism. The final column gives the part-of-speech (POS) tags for the source and target terms. We used the Penn Treebank tags (Santorini, 1990). We assigned these tags manually. Our intended mappings and our tags were chosen so that mapped terms have the same tags. For example, in A1, *sun* maps to *nucleus*, and both *sun* and *nucleus* are tagged NN. The POS tags are used in the experiments in Section 8. The POS tags are not used by LRME and they were not shown to the participants in the experiment in Section 6.

References

Ando, R. K. (2000). Latent semantic space: Iterative scaling improves precision of inter-document similarity measurement. In *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2000)*, pp. 216–223.

Banko, M., & Etzioni, O. (2007). Strategies for lifelong knowledge extraction from the web. In *Proceedings of the 4th International Conference on Knowledge Capture (K-CAP 2007)*, pp. 95–102.

Bullinaria, J., & Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, *39*(3), 510–526.

Büttcher, S., & Clarke, C. (2005). Efficiency vs. effectiveness in terabyte-scale information retrieval. In *Proceedings of the 14th Text REtrieval Conference (TREC 2005)*, Gaithersburg, MD.

Chalmers, D. J., French, R. M., & Hofstadter, D. R. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence*, *4*(3), 185–211.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science (JASIS)*, *41*(6), 391–407.

Dolan, W. B. (1995). Metaphor as an emergent property of machine-readable dictionaries. In *Proceedings of the AAAI 1995 Spring Symposium Series: Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity and Generativity*, pp. 27–32.

Evans, T. (1964). A heuristic program to solve geometric-analogy problems. In *Proceedings of the Spring Joint Computer Conference*, pp. 327–338.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*(1), 1–63.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*, pp. 1–32. Blackwell, Oxford.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2008). Cogsketch: Open-domain sketch understanding for cognitive science research and for education. In *Proceedings of the Fifth Eurographics Workshop on Sketch-Based Interfaces and Modeling*, Annecy, France.

Forbus, K. D., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. (2007). A prototype system that learns by reading simplified texts. In *AAAI Spring Symposium on Machine Reading*, Stanford University, California.

French, R. (1995). *The Subtlety of Sameness: A Theory and Computer Model of Analogy-Making*. MIT Press, Cambridge, MA.

French, R. M. (2002). The computational modeling of analogy-making. *Trends in Cognitive Sciences*, *6*(5), 200–205.

Gärdenfors, P. (2004). *Conceptual Spaces: The Geometry of Thought*. MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155–170.

Gentner, D. (1991). Language and the career of similarity. In Gelman, S., & Byrnes, J. (Eds.), *Perspectives on Thought and Language: Interrelations in Development*, pp. 225–277. Cambridge University Press.

Gentner, D. (2003). Why we're so smart. In Gentner, D., & Goldin-Meadow, S. (Eds.), *Language in Mind: Advances in the Study of Language and Thought*, pp. 195–235. MIT Press.

Gentner, D., Bowdle, B. F., Wolff, P., & Boronat, C. (2001). Metaphor is like analogy. In Gentner, D., Holyoak, K. J., & Kokinov, B. N. (Eds.), *The analogical mind: Perspectives from Cognitive Science*, pp. 199–253. MIT Press, Cambridge, MA.

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, *28*(3), 245–288.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., & Yuret, D. (2007). Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, pp. 13–18, Prague, Czech Republic.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix Computations* (Third edition). Johns Hopkins University Press, Baltimore, MD.

Hawkins, J., & Blakeslee, S. (2004). *On Intelligence*. Henry Holt.

Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database*, pp. 305–332. MIT Press.

Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 50–57, Berkeley, California.

Hofstadter, D. (2001). Epilogue: Analogy as the core of cognition. In Gentner, D., Holyoak, K. J., & Kokinov, B. N. (Eds.), *The Analogical Mind: Perspectives from Cognitive Science*, pp. 499–538. MIT Press.

Hofstadter, D., & FARG (1995). *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, New York, NY.

Holyoak, K., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, *13*, 295–355.

Holyoak, K., & Thagard, P. (1995). *Mental Leaps.* MIT Press.

Hummel, J., & Holyoak, K. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427–466.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics (ROCLING X)*, pp. 19–33, Tapei, Taiwan.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities*, *31*, 91–113.

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By.* University Of Chicago Press.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, *104*(2), 211–240.

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

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature*, *401*, 788–791.

Lepage, Y. (1998). Solving analogies on words: An algorithm. In *Proceedings of the 36th Annual Conference of the Association for Computational Linguistics*, pp. 728–735.

Lepage, Y., & Denoual, E. (2005). Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, *19*(3), 251–282.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML-98).*

Martin, J. H. (1992). Computer understanding of conventional metaphoric language. *Cognitive Science*, *16*(2), 233–270.

Marx, Z., Dagan, I., Buhmann, J., & Shamir, E. (2002). Coupled clustering: A method for detecting structural correspondence. *Journal of Machine Learning Research*, *3*, 747–780.

Mason, Z. (2004). CorMet: A computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, *30*(1), 23–44.

Minsky, M. (1986). *The Society of Mind.* Simon & Schuster, New York, NY.

Mitchell, M. (1993). *Analogy-Making as Perception: A Computer Model.* MIT Press, Cambridge, MA.

Nastase, V., & Szpakowicz, S. (2003). Exploring noun-modifier semantic relations. In *Fifth International Workshop on Computational Semantics (IWCS-5)*, pp. 285–301, Tilburg, The Netherlands.

Reitman, W. R. (1965). *Cognition and Thought: An Information Processing Approach.* John Wiley and Sons, New York, NY.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 448–453, San Mateo, CA. Morgan Kaufmann.

Rosario, B., & Hearst, M. (2001). Classifying the semantic relations in noun-compounds via a domain-specific lexical hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01)*, pp. 82–90.

Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn Treebank Project. Tech. rep., Department of Computer and Information Science, University of Pennsylvania. (3rd revision, 2nd printing).

Scholkopf, B., Smola, A. J., & Muller, K.-R. (1997). Kernel principal component analysis. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN-1997)*, pp. 583–588, Berlin.

Turney, P. D. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-01)*, pp. 491–502, Freiburg, Germany.

Turney, P. D. (2005). Measuring semantic similarity by latent relational analysis. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pp. 1136–1141, Edinburgh, Scotland.

Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics*, *32*(3), 379–416.

Turney, P. D. (2008). A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 905–912, Manchester, UK.

Turney, P. D., & Littman, M. L. (2005). Corpus-based learning of analogies and semantic relations. *Machine Learning*, *60*(1–3), 251–278.

Turney, P. D., Littman, M. L., Bigham, J., & Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pp. 482–489, Borovets, Bulgaria.

van Rijsbergen, C. J. (2004). *The Geometry of Information Retrieval*. Cambridge University Press, Cambridge, UK.

Veale, T. (2003). The analogical thesaurus. In *Proceedings of the 15th Innovative Applications of Artificial Intelligence Conference (IAAI 2003)*, pp. 137–142, Acapulco, Mexico.

Veale, T. (2004). WordNet sits the SAT: A knowledge-based approach to lexical analogy. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pp. 606–612, Valencia, Spain.

Widdows, D. (2004). *Geometry and Meaning*. Center for the Study of Language and Information, Stanford, CA.

Yan, J., & Forbus, K. D. (2005). Similarity-based qualitative simulation. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, Stresa, Italy.