JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY

DOCTORAL THESIS

Analogy as Model-Building: Exploring Analogical Inference in the Statistical Semantic Space

---

Author: Tatsuhiko Kato

Supervisor:
Dr. Shohei HIDAKA

A thesis submitted in fulfillment of the requirements for the degree of Knowledge Science

in the

Department of Knowledge Science

August 5, 2019

"... I was aware also of his thinking, like a force as palpable as heat or light or wind. This force seemed to consist in an exceptional faculty for seeing ideas as external objects and for establishing new links between ideas which appeared totally unrelated. I heard him ... treat human history as a problem in descriptive geometry, then a moment later speak of the properties of numbers in terms of zoological species. The fusion and division of cells became a particular instance of logical reasoning, and language obeyed the same laws as celestial mechanics."

"Mount Analogue", Rene Daumal (translated by Roger Shattuck)

JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY

Abstract

Department of Knowledge Science

Knowledge Science

Analogy as Model-Building: Exploring Analogical Inference in the Statistical Semantic Space

by Tatsuhiko KATO

Analogy is an ability to see the similarity between objects, in terms of the relations they share with each other. The ability has a critical importance for the study of cognition, because analogy is utilized in the broad spectrum of human activities, such as scientific discovery, metaphor generation, education and so on. However, a critical question of analogy, "where do the relations utilized to make analogy come from?", is largely ignored by the past studies. In this thesis, we take a first step to answer this question, by first providing a novel formulation of analogy and testing the formulation on the simple example. We then applied the formulation to the representative models of analogy, structure-mapping engine and word2vec, which concluded that only word2vec partially satisfies the requirements. Based on this observation, we devised a novel analogical operator for word2vec based on our formulation, which showed a decent increase in analogy performance of the model compared to previous operators. From these results, we conclude that our formulation is a promising one, and is worth further testing.

Acknowledgements

First and foremost, my deepest thanks goes to my supervisor Shohei Hidaka for his valuable advices to this thesis, as well as his teaching throughout the course of my masters period. He has helped writing this thesis in multiple ways, including suggesting the course of research, discussing ideas and reviewing the draft of the thesis. This thesis has never been in this form without him. In addition, I learned a lot from his teachings, among them I especially appreciate the attitude of careful and rigorous thinking he taught me. I do not believe I had successfully acquired this attitude even half-as-good yet, but I am determined to try practicing the attitude throughout my life.

I also thank Takuma Torii for providing several valuable advices concerning my researches, review to the draft of the thesis and being friendly in everyday life. I thank Miho Fuyama, too, who has discussed with me the important ideas and contents of the thesis, and being friendly and talkative in general. I thank my semi-supervisor Takashi Hashimoto for supporting my trip to ICCS international conference, and providing valuable inputs concerning the JSPS scholarship application. I thank Masashi Unoki for supervising my sub-project. Also, I thank all the other members of Hidaka lab, especially Shunta Koyama and Maki Miyamoto, for being friendly and wonderful human beings.

Last, but not the least, I thank all my family members, my father, mother and two sisters for their support throughout my life. I could not even pursue the academic path without their support.

The road of this masters course has been bumpy one for me, with multiple failures and disappointments with myself. I have barely made it through because of the all the supports the above (and more) people have given to me. So again, thank you all, I will do my best to apply what I learned in this period to the later part of my life.

Contents

Abstract iii

Acknowledgements v

1 Introduction 1
1.1 What is analogy, and why it's important for cognition 1
1.2 Purpose of the sutdy 2
1.3 Summary of contents 3

2 Theories and models of analogy 5
2.1 Theories of analogy 5
2.1.1 Structure-mapping theory 5
2.2 Models of analogy 8
2.2.1 Structure-Mapping Engine (SME) 9
2.2.2 Learning and Inference by Schemas and Analogy (LISA) 10
Issues with SME and LISA as a model of analogy . . . 11
2.3 Vector-space models of semantics, parallelogram theory, and word2vec 12
Parallelogram model of analogy 14
2.3.1 word2vec model 16
2.3.2 Analogy in VSMs 17
2.3.3 Possibilities and limitations of skip-gram as a model of human semantic representation 18
2.3.4 Similarity judgement 18
2.3.5 Analogy 20

3 Analogy as Model-building 23
3.1 Analogy revisited 23
3.2 Basic concepts of category theory 24
3.3 Our formulation of analogy 26
3.3.1 Explaining room analogy through the formulation . . . 26

4 Do previous models of analogy satisfy the requirements of our definition? 31
4.1 Examining SME and word2vec on whether the models satisfy the requirements of our definition 32
4.1.1 Structure-mapping engine 32
4.1.2 word2vec 32
4.2 Deriving the analogical operator from the definition 33
4.2.1 Background and Motivation 34

4.2.2 Method 34
4.2.3 Result 36
4.2.4 Discussion 38

5 Discussion and Conclusions **39**

Bibliography **41**

List of Figures

2figure.caption.8

2.1 Analogy of solar system and atomic structure (from (Gentner, 1983) with modifications). The solar system above, and the atomic structure below. S and O each stands for "Subject" and "Object". 7
2.2 Tree representation of solar-system and Rutherford-atom analogy. 9
2.3 The architecture of LISA (from (Hummel and Holyoak, 2003)) 10
2.4 Examples of one-hot vector and term-by-term matrix 13
2.5 Sumamry of VSMs learning the representation of words 14
2.6 a toy three dimensional Euclidean semantic space (from (Rumelhart and Abrahamsen, 1973), with modification) 15
2.7 Two models that consist of word2vec 16
2.8 The correlation matrix of several word embedding models and human similarity judgement (given by wordsim-353) 19
2.9 Word category-wise accuracy of analogy problems of word2vec 22

3.1 Examples of commutative diagram 25
3.2 Same figure as figure 1.1 27
3.3 a set X from the figure 3.2 28

4.1 33
4.2 Our method changes the scaling of given word vectors, so that it preserves the parallelogram relation and at the same time separate the correct word vector from noise vectors. 34
4.3 The empirical distribution of the first dimension of "noise" word vectors. 36
4.4 The result comparing the three methods, 4.2, 4.1, 4.3 with accuracy on google testset. 37

List of Tables

2.1 Overview of google testset. Category names and an example problem from the same category 20

Chapter 1

Introduction

1.1 What is analogy, and why it's important for cognition

There are two rooms depicted in the figure 1.1, one with larger size, and the specific names of furniture written, the other with smaller and black boxes in it. Although two rooms are different in terms of its size and whats in them, most likely we judge that two rooms are actually similar to each other. How do we judge that the two rooms are similar? A traditional answer from cognitive science literature might be that we can see that the shapes in the respective rooms have the same relations with each other, and that sameness of the relations leads to our judgement of the two rooms being similar. For example, Table is "on the left side of" Chair, and in the smaller room, the box which has the same shape to Table is "on the left side of" the box which has the same shape to Chair. Basically, the relation of relative placement of the boxes being "same" in the two rooms (inputs) can be thought of as the reason why people find the two rooms similar.

For us, the task of room analogy seems to be easy and nothing worth mentioning, but if we think of the task differently, it starts to seem quite difficult. For instance, think of a task as finding the "best fit" for each box and furniture based only on its shape (not on where its located in the rooms). The 7 candidates of the best fit exist for each box, and even if we assume that each box can only have one fit, there are $7! = 5040$ ways to make those fits. Among these options, we have to select one option which can be thought of as the best fit. The task only seems easy if we assume what relations we can utilize, such as "on the left side of" above. This kind of inference which utilizes "the relation of relation" (the "sameness" of the "relative placement" of boxes) is called analogy in a group of past literatures (e.g. Gentner, 1983), and has the central importance in understanding human intelligence (we call the task of finding two rooms in figure 1.1 similar "room analogy" hereafter).

Analogy has been hypothesized to be one of the hallmarks of human intelligence (Penn, Holyoak, and Povinelli, 2008), because of other animals and primates failing to perform the tasks that require the ability. For example, nonhuman animals may notice that several different apples are still same "apple" based on different apples sharing the similar color, shape, and size, i.e. perceptual features. However, only humans may notice that the relation

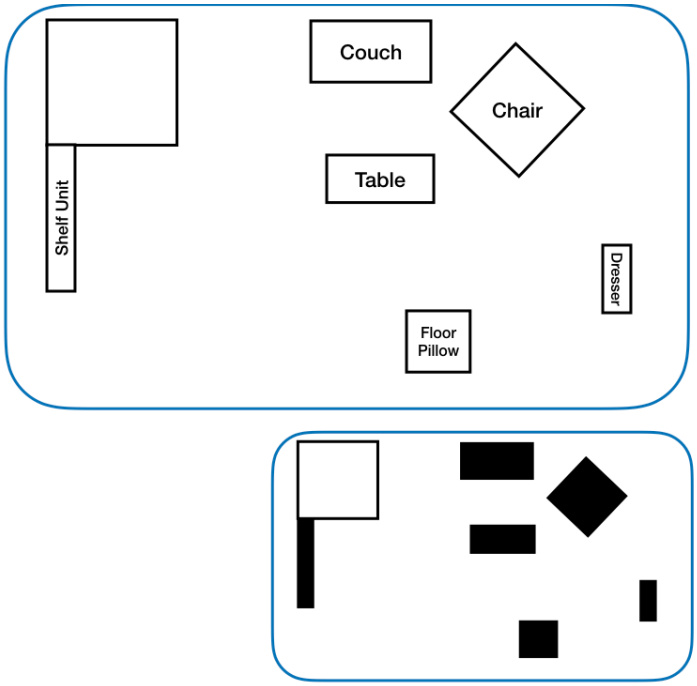


FIGURE 1.1: Even children around three years of age can find the two rooms "same" or "similar", despite the scale of objects in the rooms being different (modified from (DeLoache, 1989))

of apple and its tree is the same as the relation of child and her mother (in both cases the latter gives birth to the former).

Not only analogy might be uniquely human, but it is also seen ubiquitously in human activities, including scientific discovery (Gentner, 1983), making metaphors (Gentner et al., 2001), education, and so on. In the case of education, we can utilize analogy for connecting previously learned concepts with newly encountered ones, for example, analogy of the electric "current" and "current" of water is a popular one. The example is considered to be a good analogy because there are several properties of electrical current which correspond to properties of current of water. For instance, the difference of position or height (i.e. slope) moves the water through the pipe, creating the current. In similar vein, the difference of electrical potential (i.e. voltage) creates the electrical current.

1.2 Purpose of the sutdy

Given that analogy is important ability for humans, how can humans actually make an analogy, or in terms of what aspects are compared objects similar? As we have briefly mentioned using above, past studies have characterized analogy as the inference based on "mapping of relations between objects"(Gentner, 1983), "similarity in relational structure"(Gentner and Smith, 2013). The focus of these characterizations are clearly in the "relation".

But what is relation? and where does the relation come from? This is the central question we tackle in this paper. To better understand the significance of the question, let's go back to the room analogy example above. When we first explained the example, we just assumed we can think of the relations, such as "on the left side of" without difficulty. However, it is not clear how we can specify the relations. For example, we can think of other "relations" such as the distance in cm between boxes, or whether the colors of two boxes are same. How can we pick out a specific relation which are common in two rooms? In fact, the traditional account of analogy does not provide the answer to the latter question "Where do the relations come from?". In this paper, we present a novel formulation of analogy, which intends to answer the question above. Our formulation shifts the focus of analogy from finding same relations in the given inputs, to actually creating the inputs. By doing so, we answer the question posed above (more on this in Chapter 3).

1.3 Summary of contents

The purpose of the paper is to provide an answer to the question of "how do humans perform analogy?", especially focusing the question of "where do the relations come from?", which is largely ignored by previous researches. We tackle this task by first reviewing the relevant researches in the past on the theorization and modeling of analogy (Chapter 2), so that we can examine and point out the limitations of current theory, and later compare the theory to that of ours. Specifically, we mainly review structure-mapping theory of analogy as of theorization, and structure-mapping engine and word2vec, as of modeling. Following the review, we provide the computational formulation of analogy (Chapter 3). Then we examine the models introduced in Chapter 2, in terms of whether those models satisfy the requirements posed by our formulation. We observe that only word2vec model satisfies the important part of our formulation, namely creating the input to analogy from an unstructured data. Based on the observation, we test the formulation utilizing word2vec model, since it satisfies the important part of our requirements, by proposing the novel analogy operator for word2vec (Chapter 4).

Chapter 2

Theories and models of analogy

In this chapter, we review the past theoretical formulation and models of analogy. Specifically, we review structure-mapping theory and parallelogram theory as of theories, and structure-mapping engine (Falkenhainer, Forbus, and Gentner, 1989), Learning and Inference by Schemas and Analogy (LISA) (Hummel and Holyoak, 2003), and word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b) as of models. The core idea of former two models is based on the structure-mapping theory, with some modifications on each model. On the other hand, word2vec is more tightly connected to parallelogram theory. This review lays out the foundation for discussing our definition of analogy, and we examine the theories and models proposed here in light of our definition later in Chapter 4.

2.1 Theories of analogy

2.1.1 Structure-mapping theory

Structure-mapping theory proposed by Dedre Gentner (Gentner, 1983; Gentner, 2010) has long been regarded as the most successful theory of analogy, leading to many experimental and modeling results (Gentner and Forbus, 2011). Thus, it is important to review the structure-mapping theory to contrast later with our idea of what analogy is. The core idea of structure-mapping theory can be summarized as "analogy is about finding common relations between structured representations". The theory specifies relation as predicate, such as $SUPPORT(pillar, roof)$ and structure as the set of predicates which are also connected by second-order predicate (i.e. predicate that takes the predicates as arguments), such as $CAUSE(SHAKE, FALL)$. As can be seen from the above characterization, structure-mapping theory emphasizes the role of relation and structure of relations in defining analogy.

More formally, Gentner Gentner, 1983 defines analogy, such as "T is like B", as making the "mapping" M from a base domain B, which makes a source of knowledge, to a target domain T, which makes a domain being explained upon. Here they assume the knowledge is represented as networks of nodes and predicates. Nodes represent concepts and predicates constitute the propositions between the nodes. Then, base domain is composed of object nodes $b_1, ..., b_k$, and predicates between nodes A, R, R'. Target domain is also composed of object nodes $t_1, ..., t_k$. Mapping is defined as:

$$M : b_i \rightarrow t_i \tag{2.1}$$

The mapping M is supposed to create "the candidate set of inferences", which we would assume means a set of all relations and attributes which can be mapped from B to T, in the target domain T. This candidate set is further constrained according to the two rules:

$$A(b_i) \not\rightarrow A(t_i) \tag{2.2}$$

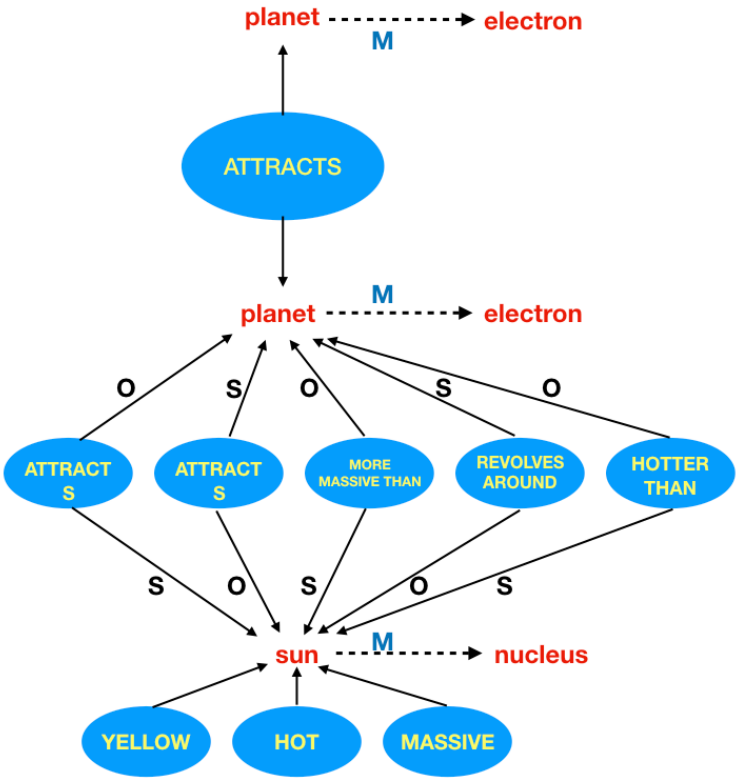
$$R(b_i, b_j) \rightarrow R(t_i, t_j) \tag{2.3}$$

Here, $\not\rightarrow$ means that the left hand side does not map to the right hand side. Predicate such as A is called "attribute", since it takes only one argument (e,g, $BLUE(earth) =$ "Earth is blue"), and predicate such as R is called "relation", since it takes two arguments (e.g. $SUPPORT(pillar, roof)$ = "Pillar supports roof"). The two rules indicate the preference of relational mapping over attributional mapping in analogy, by only mapping relations from B to T 2.3, discarding the attribute 2.2. In addition, the mapping is constrained by so called *systematicity principle*, which dictates that if there preferably map the relations which are connected by higher-order relations (i.e. relations which take predicates, not objects, as arguments). This constraint is formally defined as:

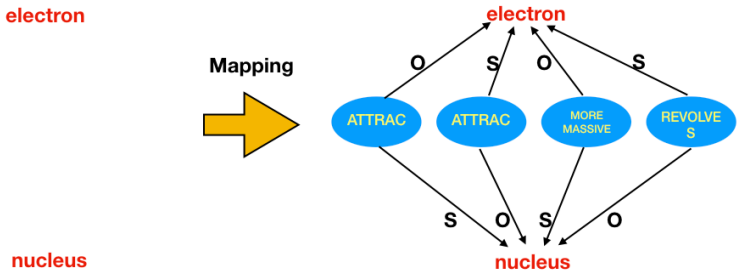
$$R'[R(b_i, b_j), R(b_k, b_l)] \rightarrow R'[R(t_i, t_j), R(t_k, t_l)] \tag{2.4}$$

that This process of creating M and candidate set of inferences from M, then cutting down the candidate set, constitutes the structure-mapping theory of analogy. To sum up, the theory postulates that analogy is to define a mapping M from a base B to a target T. To create a mapping, first create the mapping from base objects to target objects (2.1), then only map the relations, but not the attributes, from the base (2.2 and 2.3)[1]. In addition, choose the relations which have the higher-order relation between them (2.4). Next, we examine a concrete example using the theory, and see how the formulation can be utilized to understand the process of making an analogy.

[1] In fact, Gentner is ambiguous on how much attribute should be mapped. In some parts she writes "Discard attributes of objects", and in some other parts "few or no object attributes" can be mapped.



(A) Base domain of Rutherfold analogy.



(B) Target domain of Rutherfold analogy, and the mapped relations between objects.

FIGURE 2.1: Analogy of solar system and atomic structure (from (Gentner, 1983) with modifications). The solar system above, and the atomic structure below. S and O each stands for "Subject" and "Object".

An example analogy is the one which is used by the physicist Ernest Rutherford, when he came up with his model of atomic structure by finding a similar structure between solar system and hydrogen atom (Encyclopaedia Britannica, 2018). Given the kind of "structured representations" shown in figure 2.1, structure-mapping theory specifies how the mapping and candidate set of inferences should be created from the base (solar system) to the target domain (atomic structure).

Figure 2.1 shows the representations of base and target domains. Referring to the previous section, according to (Gentner, 1983), making an analogy is to define a mapping M from base (solar system) to target (atomic structure). Then from the candidate set of inferences generated from M, we choose the

one which only maps the relations, but not the object attributes. Thus in this example, only ATTRCTS(S, O), ATTRCTS(O, S), MORE MASSIVE THAN(S, O), and REVOLVES AROUND(O, S) are mapped and other object attributes, such as YELLOW(sun), are discarded. The relation HOTTER THAN(S, O) is not mapped, because of the systematicity principle 2.4. Specifically, there is no systematic relation over HOTTER THAN(S, O), unlike other relations. They argue that other relations than HOTTER THAN(S, O) are systematically related, because changing one relation among them affects the other relations. Think of the four relations summarized below:

1. DISTANCE(sun, planet)
2. ATTRACTIVE FORCE(sun, planet)
3. REVOLVES AROUND(planet, sun)
4. MORE MASSIVE THAN(sun, planet)

(Among the four, DISTANCE(sun, planet) is not depicted in the figure 2.1 and ATTRACTIVE FORCE(sun, planet) is divided into two relations ATTRACTS(S, O) and ATTRACTS(O, S)). Changing the (1) distance between sun and planet also changes the (2) attractive force between sun and planet. Also changing (4) to MORE MASSIVE THAN(planet, sun), changes (3) to REVOLVES AROUND(sun, planet). Thus these relations are systematically related, but HOTTER THAN(S, O) is not, which is why HOTTER THAN(S, O) is not mapped in the target domain. Note that Gentner, 1983 does not specify how can we make the object mappings M in the example. For instance, why is it the case that $planet \rightarrow electron$, but not $electron \rightarrow planet$?

Currently, structure-mapping is the representative theory of analogy and other theories are more or less the variants of it. For instance, multiple-constraint theory of analogy by (Holyoak and Thagard, 1989), adds several other constraints for analogy other than *systematicity* of relations. These include how similar the words in domains are (i.e. semantic constraint), and the purpose of the person making analogy (i.e. pragmatic constraint). Thus, although there are several attempts to theorize analogy and minor modifications on each of them, the idea of structure-mapping (i.e. the importance of finding relational correspondences and creation of mapping afterwards) has been central in almost all of the theoretical formulation (Holyoak, Gentner, and Kokinov, 2001).

2.2 Models of analogy

In the last section, we have described structure-mapping theory as the representative theory of analogy. Other than experimental predictions that the theory has made, the reason that the theory has been widely accepted among the researches is that they created a working computational model, "structure-mapping engine" (SME), based on the idea of the theory. The multitudes of the models have been created since then (see (Gentner and Forbus, 2011) for

the overview of the models), and although we can not review all of them here, we will briefly take a look at the two widely accepted models: SME and LISA. We review the models here to later judge whether these models satisfy our definition of analogy.

2.2.1 Structure-Mapping Engine (SME)

Structure-mapping engine was proposed as an implementation model of structure-mapping theory (Falkenhainer, Forbus, and Gentner, 1989). The model takes as input the base and target domains which are structured as directed acyclic graph (DAG) which take predicates as nodes, and argument relations (whether some predicates are arguments of other predicates) as edges. DAG is a graph on that there is a path from a node to another node for only one direction, and if we start following paths from a node, we can not go back to the same node again (an example of such graph is shown on figure 2.2). SME makes an analogy by finding a mapping and making inferences over those DAG inputs.

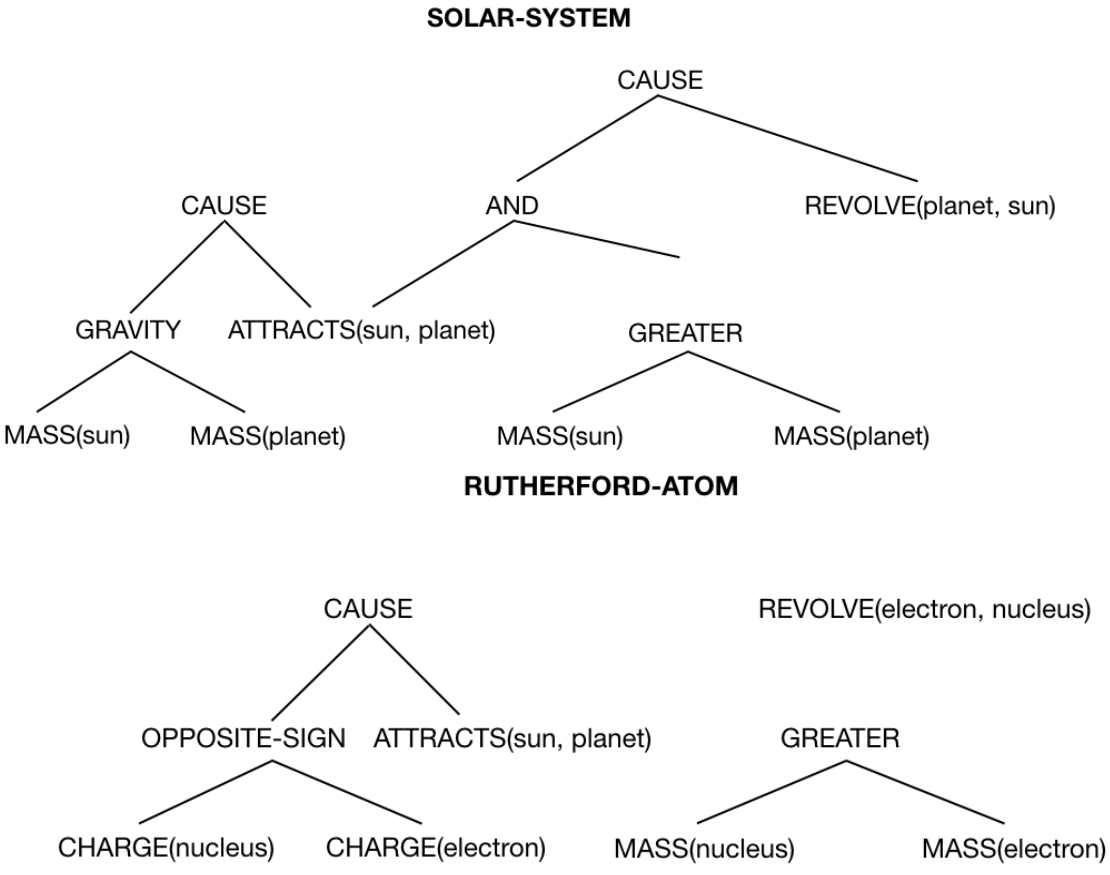


FIGURE 2.2: Tree representation of solar-system and Rutherford-atom analogy.

In the mapping process, the model proceeds in four steps:

1. Construct the all potential matchings among entities and relations

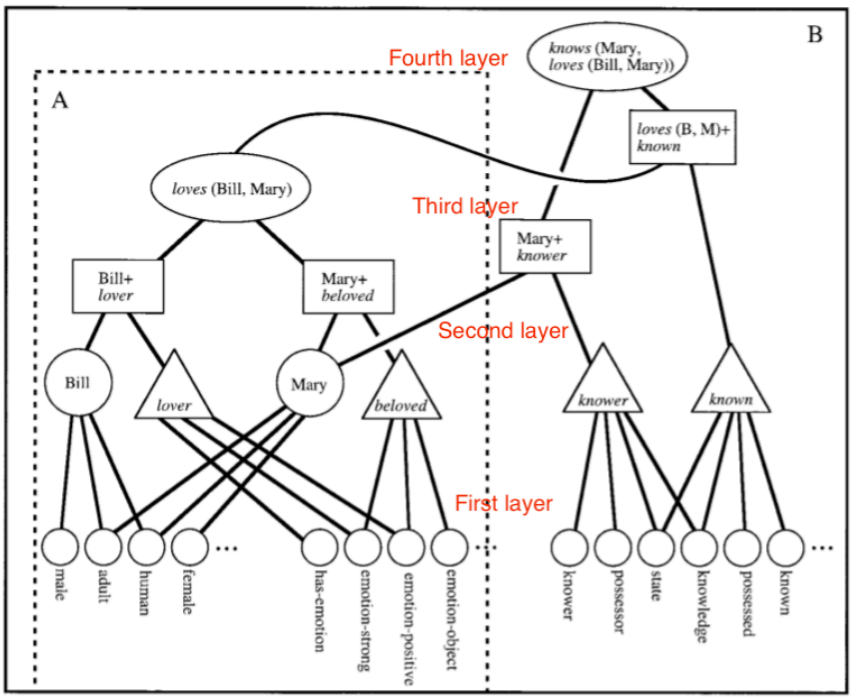


FIGURE 2.3: The architecture of LISA (from (Hummel and Holyoak, 2003))

2. Construct possible mappings by combining the matchings
3. Compute candidate inferences from mappings
4. Evaluate matches hypotheses based on the evidence

The first part of the process constructs the matchings of relations based on the node labels. If two relation nodes share the same label, the model creates a match. After constructed the possible matchings, the model proceeds to construct mappings by combining the structurally consistent matchings. Structurally consistent matchings refer to those matchings that share the same root node in the tree structure, discarding any possible matchings which are not connected by relations, such as object attribute. This process reflects the three rules in SMT (2.2, 2.3). After the mapping construction, the model computes the possible inferences based on the mapping. Here, inference refers to the process of complementing target domain with any nodes that are present in the base mappings but not in target mappings. Then after the three steps, the model assigns the scores to each matchings based on whether the matchings share the higher-order predicate, reflecting the rule three 2.4, of SMT.

2.2.2 Learning and Inference by Schemas and Analogy (LISA)

LISA (Hummel and Holyoak, 2003) is a model of analogy-making based on the idea of multiple constraint theory, and is implemented on neural network architecture. The basic architecture of LISA is shown on the Figure 2.3. Let us explain the basic assumptions the modelers take using the figure.

The basic idea of human analogy-making employed in LISA is that predicates, which are employed in the relational matchings (as in SME), are represented as a combination of roles, fillers and their bindings with each other. Roles are characteristics of objects they display in certain situations. In Figure 2.3, lover, beloved, knower, and known are roles. Fillers are objects that "fill" the roles as defined above. Bill and Mary are fillers in this sense. Roles and fillers are bound together to form larger units, as in "Bill+lover" in the figure.

The architecture of LISA reflects the above assumption that humans make relational matchings on roles, fillers and bindings. Figure 2.3 shows the overall architecture of LISA. There are four layers, starting from the bottom, the first layer corresponds to semantic units, which specify the attributes that an object posses. In the figure, Bill is connected to three semantic units, "male", "adult", and "human", which characterize this object "Bill". The second layer contains objects and relations, such as "Bill" or "lover". The third layer dictates the bindings between objects and relations. Bindings in turn make up the "sub-proposition" as in "loves(Bill, Mary)" in the figure. This is called sub-proposition, because they assume the block A in the figure in the whole represents the proposition, not "loves(Bill, Mary)" only. With this architecture, analogical mapping is discovered based on what units co-activate with what other units in the input representation.

Issues with SME and LISA as a model of analogy

In the last two sections, we have reviewed two models, structure-mapping engine (SME) and Learning and Inference by Schemas and Analogy (LISA). In this section, we argue that both models suffer from the same problem, namely the arbitrariness of the making of input domains. Possibly the most serious criticism towards SMT and SME (and other variants of it, including LISA) is the one raised by (Chalmers, French, and Hofstadter, 1992). The main criticism of (Chalmers, French, and Hofstadter, 1992) is that there seems to be so much arbitrary choice made by researchers in SME, that it makes the problem of analogy-making trivial. Take the analogy problem depicted in figure 2.2 as an example. If you just make correspondences of the nodes sharing same labels, it is pretty easy to find the mapping from sun to nucleus, and from planet to electron, even for computer programs. This is because, so they point out, the representation of the domains is built by researchers to solve the specific analogy in mind. To quote from (Chalmers, French, and Hofstadter, 1992): "Since the representations are tailored (perhaps unconsciously) to the problem at hand, it is hardly surprising that the correct structural correspondences are not difficult to find."

Another arbitrariness of SME, they argue, is the use of the words "object", "attribute", and "relation". They claim that, although in the input to the SME it is fixed whether some concept is object, attribute or relation, in human mind, what things are object, attribute or relation might change flexibly. For example, "wealth" can be an object that flows from one agent, or can be an attribute associated with the agent that changes with each transaction. We

suppose that the same criticisms as above two can be applied to LISA and many other models, since the input to those models are already constructed representations of the domains.

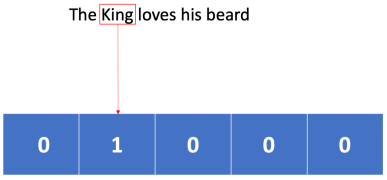
We agree with Chalmers, French, and Hofstadter, 1992 in that the disregard of representation-building by analogy models is problematic, not only because it makes the process of analogy-making trivial, but also it indicates that SMT is incomplete as a theory of analogy. This is because, if SMT is complete as a "theory" of analogy, it must specify the necessary and sufficient conditions for some inference to be analogy. In other words, the inability of SME to construct the suitable representation for analogy indicates that SMT, which SME is based on, is at least not specific enough to give the model an ability to solve analogy by itself.

In the next section, we see that a group of models, called vector-space models, proposed in natural language processing partially solves this problem of inability for analogy models to construct representation by itself.

2.3 Vector-space models of semantics, parallelogram theory, and word2vec

In this section, we introduce vector-space models of semantics (VSMs) in general, and then word2vec model, which is a representative model of VSMs, in detail, focusing on the the connection between the models and human semantic representation. The significance of VSMs for the modeling of analogy lies in the fact that, first, recently some VSMs have demonstrated that the models can solve several types of four-term analogy problems via vector operations (Mikolov et al., 2013a). Four-term analogy problem is a type of analogy problem, which presents the solver three words, say "Man", "King", and "Woman", and then let the solver find the fourth word, which holds the same relation to "Woman", as "King" to "Man". This type of problems is often presented as "Man : King :: Woman : ?". The four-term analogy problem is widely used in analogy research as well, since being able to solve the problem is considered to reflect the basic ability of analogy (Holyoak, Holyoak, and Thagard, 1995, chapter 2 and 3). Second, as we have briefly mentioned in the previous section, VSMs can overcome the arbitrariness that traditional analogy models suffer, by constructing the representation of the domains of analogy ground up from the corpus data. These two are the reasons we review the VSMs as models of analogy.

VSMs (Turney and Pantel, 2010) refer to the models which typically take the word vectors represented as one-hot vector 2.4a as input, and embed the one-hot vectors in the continuous space, preserving the relationships among words, such as term-term cooccurrence 2.4b, present in the corpus. Term-term cooccurrence refers to the matrix which each row and column is some word in the corpus, and the values of the matrix correspond to how many times words in the row and column occurred in some range of words. One-hot vector is a vector which represents words in the corpus, making the only value in the vector on which the relative position of the word appearing in



(A) An example one-hot vector of the word "king"

	King	Love	Crown	Beard	Walk	Man	Queen	Cosmetics	Woman
King	1	1	1	1	1	1	1	0	1
Love	1	1	0	1	1	1	1	0	1
Crown	1	0	1	1	0	1	1	0	0
Beard	1	1	1	1	0	1	0	0	0
Walk	1	0	0	0	1	1	1	0	1
Man	1	1	0	1	1	1	1	0	1
Queen	1	1	1	0	1	1	1	1	1
Cosmetics	0	0	0	0	0	0	1	1	1
Woman	1	1	0	0	1	1	1	1	1

(B) An example term-by-term cooccurrence matrix of the 10 words in figure 2.5

FIGURE 2.4: Examples of one-hot vector and term-by-term matrix

the corpus to 1. For example in figure 2.4a, the word "king" is represented as the vector $[0,1,0,0,0]$, since the word "king" appears in the second place of the sentence. The problem with representing words as one-hot vectors is that it is hard to compute the relationship of words in a meaningful manner. For instance, if we compute the cosine similarity $\cos(v,w) = \frac{v^\top w}{\|v\|\|w\|} = \frac{\sum_i v_i w_i}{\sqrt{v_i^2}\sqrt{w_i^2}}$ (v, w are vectors and v_i, w_i are components of vectors, $\top$ is transpose of vectors, and $\|v\|$ is norm of a vector) on any different one-hot vectors, it simply gives the value 0, since $\sum_i v_i w_i$ always gives the value 0 on different one-hot vectors (i.e. vectors are orthogonal). VSMs solves this problem by constructing the vectors which have continuous values on each dimension, unlike one-hot vector, making the vectors non-orthogonal. This ability of making relationships of word vectors easily computable, is why VSMs is widely utilized in natural language processing (NLP) field. Computing the relationships of word vectors is important for the field, since to conduct any task in NLP, such as sentiment analysis[2] or machine translation, one needs some kind of similarity measure between linguistic entities, such as words, sentences, or documents.

VSMs is fundamentally based on the idea that cooccurrence of words in the corpus provides a good approximation to the meaning of words. The idea is known as "distributional hypothesis". The idea of the hypothesis can be summarized as "you shall know the meaning of words by the company it keeps" (Firth, 1957). For example (2.5), the word "King" may cooccur with the words "Love", "Crown", or "Beard". On the other hand, the word "Queen" may cooccur with the words "Love", "Crown", or "Cosmetics". Thus, if you quantify the similarity of the words in terms of what other words the words cooccur with, "King" and "Queen" are pretty similar, although not exactly the same. By gathering this kind of information for other words as well, you get a measure of how similar the words are. VSMs implements the idea through, typically, making the distance between word vectors in the space reflect cooccurrence statistics among words in the corpus.

[2]The task is to classify the words, sentences or documents based on the primary emotional element, such as happiness or anger, of the content.

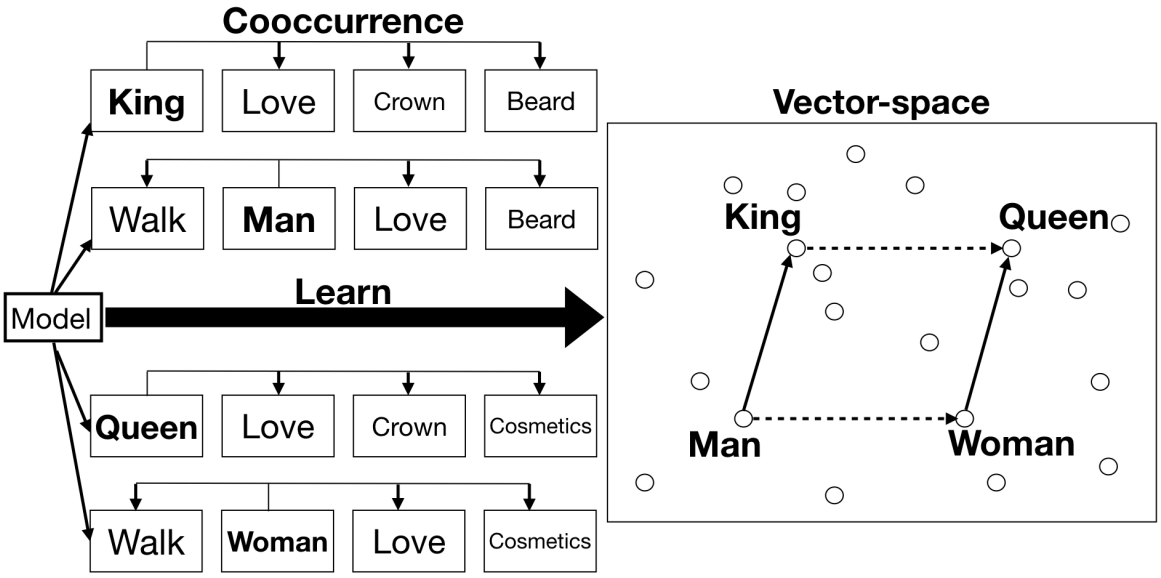


FIGURE 2.5: Sumamry of VSMs learning the representation of words

Parallelogram model of analogy

The model of analogy which holds a strong connection to VSMs is parallelogram model. SMT emphasized the role of predicates and symbolic representations in defining relation and analogy, on the other hand, parallelogram model Rumelhart and Abrahamsen, 1973 formulates analogy in terms of the vector operations in Euclidean space, as we discuss further soon. Although their theoretical formulation has not been popular in analogy literature for decades, the formulation is recently revived through the advent of word embedding models, such as word2vec (Chen, Peterson, and Griffiths, 2016). This is because word embedding models utilize the nearly identical assumptions and vector operations to implement analogy as what Rumelhart and Abrahamsen proposed. Thus, here we review the parallelogram model before actually take a look at word2vec model.

In their formulation, Rumelhart and Abrahamsen, 1973 emphasized that to perform any similarity judgement, the following two questions need to be answered:

1. what is the nature of the memory structure which underlies similarity judgements?

2. what is the measure of ”distance” on this psychological space?

Here, memory structure refers to the way our concepts are represented in the brain. According to them, Henley, 1969 provides a set of answers to the questions:

1. the memory structure may be represented as a multidimensional Euclidean space.

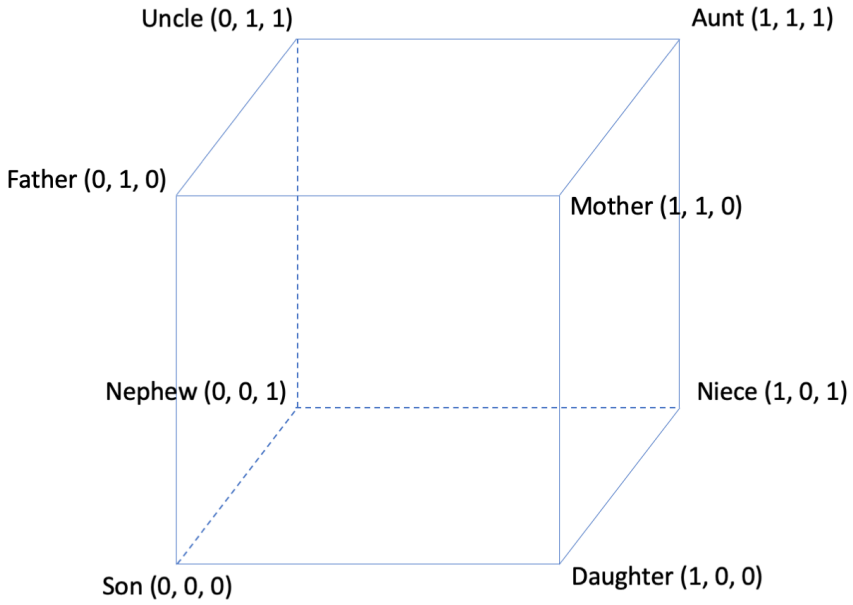


FIGURE 2.6: a toy three dimensional Euclidean semantic space (from (Rumelhart and Abrahamsen, 1973), with modification)

2. judged similarity is inversely related to distance in this multidimensional space.

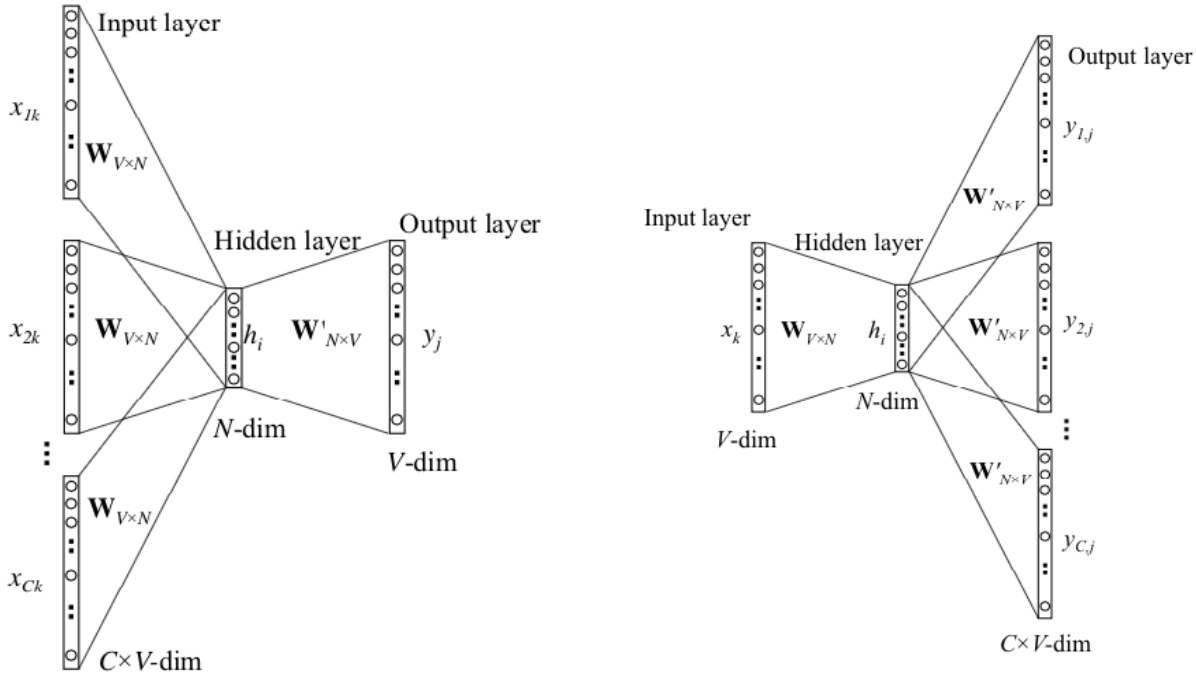
Rumelhart and Abrahamsen also accept the answers. Based on these assumptions, they proceed to claim that analogical reasoning can also be seen as a kind of similarity judgement, but the one in which "not only the magnitude of the distance but also the direction must be indicated" (i.e. vector distancec). When people perform an analogy like A is to B as C is to D, they argue, people simply state that A is similar to B in "exactly the same way and exactly the same degree" (Rumelhart and Abrahamsen, 1973) as C is similar to D. More formally, for any analogy problems A:B::C:?, there exists a concept I corresponding to the answer of "?", such that I is located at the same distance from C as B is from A. The coordinates of I are specified by the operation on the ordered sequence ,

$$\{c_j - a_j + b_j\}_{j=1:m}. \tag{2.5}$$

Here, c_j, a_j, b_j are the elements of vectors, and m is the dimension of the vectors. As we see later, this vector operation 2.5 which utilizes the addition and subtraction on the word vectors, is identical to that of used by the paper which proposed word2vec (Mikolov et al., 2013a).

To illustrate the effectiveness of their formulation, let's take a look at the simple example in the paper. Figure 2.6 represents a toy three dimensional Euclidean space, in which each point corresponds to some concept. Here, eight concepts related to family are selected as an example. To solve an analogy problem, for instance Son:Daughter::Father:?, utilizing the algorithm Rumelhart and Abrahamsen, 1973 proposed, we just need to calculate

$$\{Father_j - Son_j + Daughter_j\}_{j=1:3} = (0-0+1, 1-0+0, 0-0+0) = (1,1,0)$$



(A) CBOW model architecture

(B) Skip-gram model architecturre

FIGURE 2.7: Two models that consist of word2vec

and this corresponds to Mother. Ofcourse, this is highly simplified and idealized example, and parallelogram theory was forgotten in part because it couldn't be applied to more complex examples (Chen, Peterson, and Griffiths, 2016). However, later we will see that recent word embedding models can solve much more varieties of analogy problems through the "memory structure" constructed from large amount of corpus, using the similar idea introduced here.

2.3.1 word2vec model

Currently, one of the most widely used models of VSMs is word2vec (Young et al., 2017). Word2vec consists of two models, continuous bag-of-words (CBOW) and skip-gram. Both models are a single-layer (meaning the network has only one hidden layer) neural network, and skip-gram model takes one N dimensional word vector (represented as one-hot vector) as input and tries to maximize the prediction accuracy in output layer for the C words around the input word in the corpus (figure 2.7a). CBOW model does the opposite by taking C number of N dimensional word vectors and maximize the prediction accuracy in output layer for a single target word (figure 2.7b). In figure 2.8, x_ik, x_k is an input one-hot vector, $\mathbb{W}_V$, $\mathbb{W}'_N$ are weights of the network corresponding to each input vector. Each input vector is multiplied by weight matrices to produce an output vector. Since skip-gram generally has better performance on analogy tasks introduced later (Mikolov et al., 2013a) and is utilized more commonly, here after we focus on skip-gram model. The model optimizes vector representation x_k so that it maximizes the conditional

probability of $t^{\text{th}} + k$ word given the other words occurring within the time window of size c:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log P(v_{t+j}|v_t),$$

where the the probability function P is given by the soft-max function $\frac{e^{x_i}}{\sum_i e_{x_i}}$.

2.3.2 Analogy in VSMs

In the last two sections, we have introduced VSMs and skip-gram model. A reason skip-gram model has gained popularity among general populations, in addition to researchers, is its ability to solve four-term analogy problems such as "Man : Woman :: King : ?". This section describes how skip-gram model solves this kind of problems. The four-term analogical inference task in the context of VSMs, has been introduced by the same paper as the one which proposed CBOW and skip-gram, which is (Mikolov et al., 2013a), and has become a standard benchmark test for the word representation models. In the four-term analogy task, a model is given a triplet of words in the form $(Man : Woman :: King : x)$, and is asked to answer the missing fourth word x. The model answers the fourth word by applying some similarity operator to the given triplet of the three words for each question. We call this process solving the four-term analogy problems with VSMs. A straightforward method for solving the analogical inference task is the one given by (Mikolov et al., 2013a; Mikolov et al., 2013b).

$$f(v_a, v_b, v_c) = \underset{v_d\notin\{v_a,v_b,v_c\}}{\arg\max}\ (\cos(v_c - v_a + v_b, v_d)) \tag{2.6}$$

arg max refers to the operation which takes a vector as input and returns the dimension of the highest value in the vector. The method takes the offset of given word vectors and find the word vector which has the highest cosine similarity to the offset term. Levy and Goldberg, 2014 proposed another method to solve the four-term analogy task:

$$f(v_a, v_b, v_c) = \underset{v_d\notin\{v_a,v_b,v_c\}}{\arg\max}\ \frac{\cos(v_b, v_d)\cos(v_c, v_d)}{\cos(v_a, v_d)} \tag{2.7}$$

The method 2.7 has currently the best accuracy in analogy tasks (Linzen, 2016). Both methods are based on the idea that to get the desired result for four-term analogy task, make the v_c vector as far, in terms of the distance, from v_a and as near to the v_b. Using the same example as above, make the v_{king} vector as far from v_{man} and as near to v_{woman}. We can easily see the similarity between 2.6 and the one proposed by (Rumelhart and Abrahamsen, 1973), 2.5, except for in 2.6 cosine similarity is used to measure the similarity of the vectors. Both methods try to capture the relationship between four word vectors, which can be represented as in figure 2.5 (parallelogram relationship (Chen, Peterson, and Griffiths, 2016; Rumelhart and Abrahamsen, 1973)). This parallelogram relationship can be understood as the two pair of

word vectors having similar relations (i.e. distance in the vector-space) with each other, which indicates that the model captures an important aspect of word relationship in the corpus. This interpretation of the model capturing consistent relations between the four words is the reason that analogy task is utilized to test the model's performance on representing meanings of words in NLP.

2.3.3 Possibilities and limitations of skip-gram as a model of human semantic representation

In addition to solving analogy problems, skip-gram model is also shown to be able to achieve high correlation with semantic similarity judgement of humans (Baroni, Dinu, and Kruszewski, 2014). In this section, we review the related literature and present some of our own analysis, regarding how well the word2vec model can capture human semantic judgement and analogy ability.

2.3.4 Similarity judgement

Similarity judgement is the task that makes humans or models to decide the similarity of the pairs of words. Typical experimental procedure for humans provides the pairs of words to the participants and let them decide the similarity scores in the specified range (from 0 (not at all similar)-10 (very similar), in the case of WordSim-353 (Finkelstein et al., 2002) and SimLex-999 (Hill, Reichart, and Korhonen, 2014)). In the case of the model, the model decides the similarity of pairs of words using cosine similarity already mentioned. There are several open datasets of similarity judgement, which include the word pairs and the scores the participants of the experiment assigned to the pairs. The well-known datasets are WordSim-353 (Finkelstein et al., 2002), MEN (Bruni, Tran, and Baroni, 2014), and SimLex-999 (Hill, Reichart, and Korhonen, 2014). By using several similarity datasets, (Baroni, Dinu, and Kruszewski, 2014) tested the similarity judgement performance of CBOW model, by taking the Pearson or Spearman correlation of the model (cosine similarity) and human similarity judgement scores of word pairs. The model achieved the high correlation scores across datasets, with $r = 0.75$ for WordSim-353, $r = 0.8$ for MEN (r being Spearman correlation).

The above results by (Baroni, Dinu, and Kruszewski, 2014) show the impressive ability of word2vec to imitate human similarity judgement. We partially replicated the correlation result for WordSim-353 and further compared the correlation results of models to that of each participant's similarity judgement scores of word pairs (figure 2.8). As the models, besides word2vec (skip-gram) already introduced, we utilized GloVe (Pennington, Socher, and Manning, 2014) which is a VSMs model similar to word2vec, and ConceptNet 5 (Speer and Havasi, 2012), which also is a similar model to word2vec, but also utilizes the external hand-coded knowledge. As readers can see in figure 2.8, human-human ($mean = 0.723$) and model-model

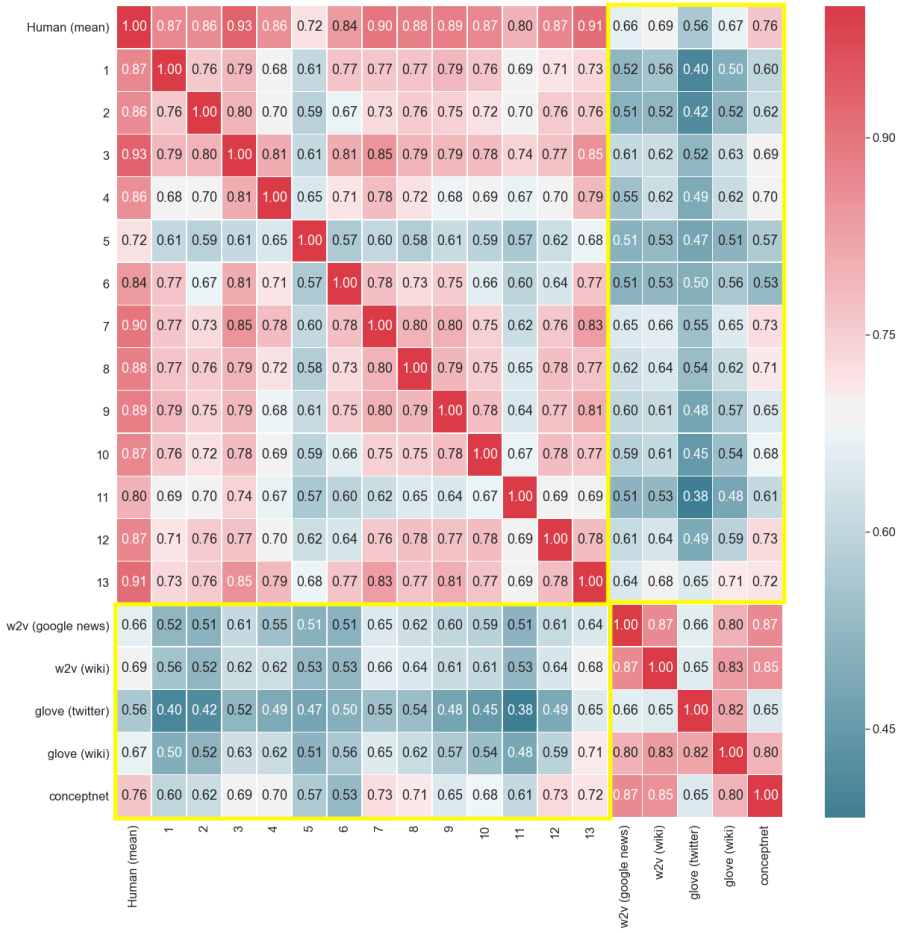


FIGURE 2.8: The correlation matrix of several word embedding models and human similarity judgement (given by wordsim-353)

Category names in the testset	Example problem
capital-common-countries	Berlin : Germany :: Paris : France
capital-world	London : England :: Rome : Italy
currency	Japan : yen :: USA : dollar
city-in-state	Boston : Massachusetts :: Honolulu : Hawaii
family	man : woman :: king : queen
gram1-adjective-to-adverb	amazing : amazingly :: calm : calmly
gram2-opposite	acceptable : unacceptable :: aware : unaware
gram3-comparative	bad : worse :: big : bigger
gram4-superlative	bad : worst :: big : biggest
gram5-present-participle	code : coding :: dance : dancing
gram6-nationality-adjective	France : French :: Germany : German
gram7-past-tense	dancing : danced :: decreasing : decreased
gram8-plural	banana : bananas :: bird : birds
gram9-plural-verbs	decrease : decreases :: describe : describes

TABLE 2.1: Overview of google testset. Category names and an example problem from the same category

($mean = 0.772$) correlation scores are consistently higher than the model-human ($mean = 0.558$) correlation scores (scores blocked by yellow lines). This result of the model-human correlation being pretty lower than human-human and model-model correlation indicates that, although word2vec can represent the human similarity judgement pretty well, there still is a gap between humans and models, even in this kind of relatively simple task of word pair similarity judgement.

2.3.5 Analogy

As mentioned in the section 2.2.5, four-term analogy problem is introduced in (Mikolov et al., 2013a), and has become a standard test for semantic representation ability. The standard testset for analogical inference, which is also proposed in the same paper, is known as google test set (Mikolov et al., 2013c). The test set consists of two large categories, namely semantic and syntactic category, which are further categorized into fourteen categories which reflect their choice of analogy questions (table 2.1). As can be seen from the table 2.1, analogy problems utilized in this set are rather simple, in the sense that at least adult humans should have little trouble coming up with the target word.

Using the google test set, (Mikolov et al., 2013a) reports the overall accuracy (i.e. the number of correctly answered questions divided by the total number of questions) of 53.3% using skip-gram, which is 20-30% higher than other models with same amount of training words, and even higher than the other model which used 10 times more amount of words for training than skip-gram. The utilized models solved the problems utilizing the method 2.6. The results show skip-gram's significant improvement of accuracy on four-term analogy problem over other models.

We analyzed the performance of skip-gram model on the same google test set, but focusing on the accuracy of each problem categories on the test set 2.1, not on the overall accuracy, to see the models performance in more detail. The figure 2.9 shows the result of category-wise accuracy of the model, using the same google test set, and pre-trained skip-gram vectors Mikolov et al., 2013d, which is trained on about 100 billion words in the articles of Google News. As (Mikolov et al., 2013a), we utilized the method 2.6 to solve the problems on google test set. The figure 2.1 shows that, although skip-gram can solve the problems fairly well (more than 60% accuracy) in most of the categories, there is a clear drop of performance (around 20%) on the three categories, namely currency, gram1-adjective-to-adverb, and gram2-opposite. As we have noted earlier, the questions in the test set themselves are fairly easy ones for humans to answer. The result points to the discrepancy of the humans and models semantic representation ability. To amend the discrepancy, In chapter 4, we propose a new method for analogical inference based on our definition of analogy, which improves the performance, especially on the three categories.

To sum up, in this chapter we introduced and reviewed the theory and models of analogy. Strucure-mapping theory is currently the most important theory of analogy, which considers analogy as finding the mapping from the base to the target, making the mapping only on the relations both in the base and the target. As of models, we have reviewed structure-mapping engine and word2vec. SME is the model of analogy, which implements the idea of structure-mapping, and word2vec is the model of semantic representation, on which some analogy problems can be solved. Most importantly, following (Chalmers, French, and Hofstadter, 1992), we have pointed out the limitations of SMT and SME, namely the underspecification of what can be considered as the input to analogy, and arbitrariness of the input domains of the model. Word2vec partially overcomes the problem of SME pointed out here, by creating the representation of domains in analogy from the corpus without man-made input, but can only solve simple four-term analogy problems compared to SME, which can solve larger scale problems (although its debatable whether SME can even be said to solve the *simplest* of analogy problems, since it needs a meticulous human intervention).

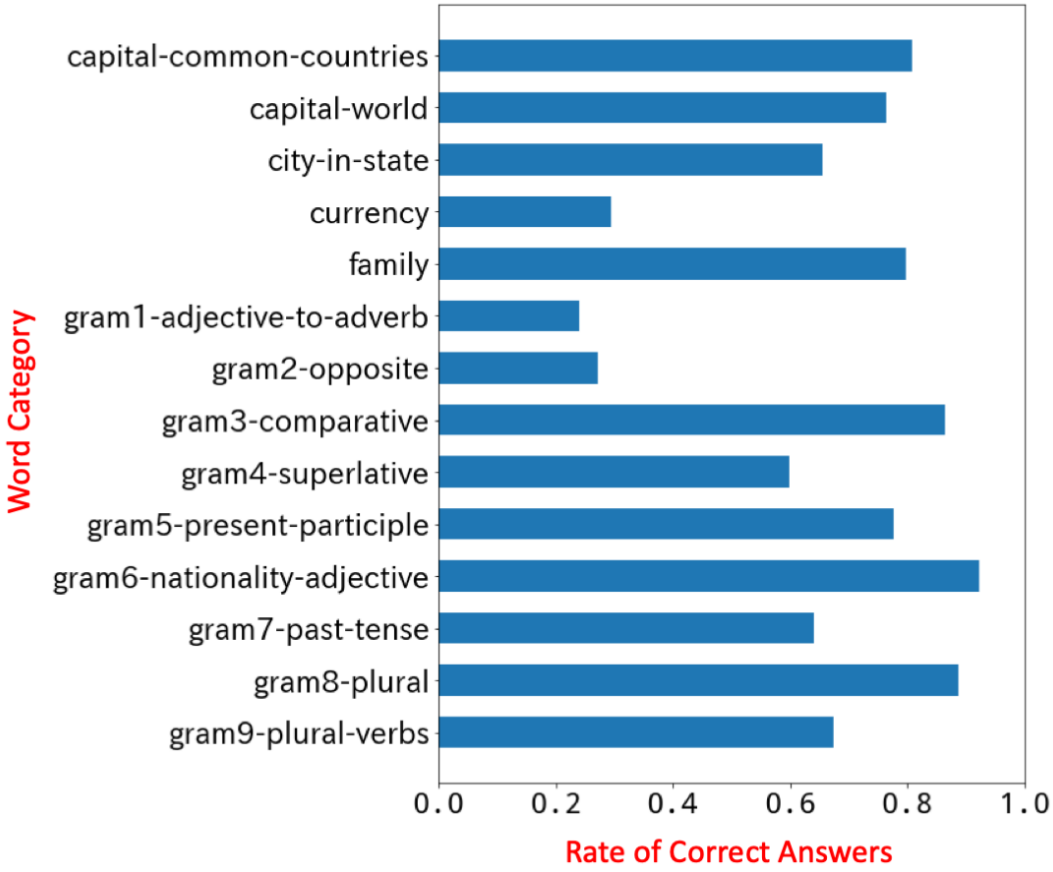


FIGURE 2.9: Word category-wise accuracy of analogy problems of word2vec

Chapter 3

Analogy as Model-building

In the last chapter, we have looked at past theoretical formulations and models of analogy. In this chapter, we develop our own formulation of analogy. First, we hypothesize that analogy is to construct a representation of "entities" and "relations" between entities, so that the entities satisfy a certain condition. Here, entities refer to something that can be considered as a whole and relations refer to functions. We call this process "model-building" as in the chapter title, since the process involves creating a representation of something to be modeled, in terms of entities and relations under a certain condition. To formulate the idea, we utilize category theory, because it provides us with the tools to concisely organize the idea of our hypothesis. Category theory is a field of mathematics that treats the relations between mathematical entities. The basic concepts in the theory are defined in terms of what mathematical relations they hold to other concepts in the theory.

3.1 Analogy revisited

In this section, we describe our hypothesis of analogy, and why we consider category theory a potentially useful tool to formulate our idea of analogy. As is discussed in Chapter 1, analogy has been considered to have a deep connection with concept of "relation" and "relation of relation". We align with previous researches as of the importance of relation in analogy, but contend that we should also include the definition on the domains in the theory. Thus, we hypothesize that given a set X of $x_1, x_2, ..., x_n$, analogy is to:.

1. from X, construct quadruplets of sets $< A, B, R_A, R_B >$ so that two functions $f : A \to B, F : R_A \to R_B$ satisfy $F(g) \circ f = f \circ g$
2. then select a quadruplet $< A, B, R_A, R_B >$ which have largest number of elements

Here, $a_1 = x_1, a_2 = x_2, ..., x_i = a_i \in A, b_1 = x_{i+1}, b_2 = x_{i+2}, ..., x_{i+k} = b_i \in B$ are the sets which consist of the elements of X, where $A \cap B = \emptyset$, and R_A, R_B are the sets of functions , for any element of $a_i, a_j \in A, b_i, b_j \in B$, $h_A : a_i \mapsto a_j \in R_A, h_B : b_i \mapsto b_j \in R_B$, each consisting of elements of A, B. Condition 2 is necessary because there can be infinitely many combinations of quadruplets $< A, B, R_A, R_B >$. In this hypothesis, R_A, R_B correspond to "relation", and the constraint of $F(g) \circ f = f \circ g$ corresponds to "relation of

relation". Hence, our hypothesis defines analogy as constructing a representation of entities A, B and sets of relations R_A, R_B, under the constraint of relation of relation $F(g) \circ f = f \circ g$. To contrast with structure-mapping theory (SMT), SMT defined analogy as finding a mapping from a given base domain, which includes entities and relations structured as directed acyclic graph, to a given target domain, which also includes entities and relations. The emphasis of our definition lies in the role of *constructing* the domains of analogy. In SMT, the construction part is completely ignored as the focus is largely put on how the mapping should be created. However, as it is pointed out by (Chalmers, French, and Hofstadter, 1992), this disregard of "representation-building" leads to the underspecification of the theory, which then leads to the arbitrariness of the model based on the theory. All we assume is an input of a set X, which only minimally assumes that something can be composed of its elements, unlike SMT.

In this section, We have taken a first step to define analogy. However, our definition so far is lacking a formality which is required to judge whether the model really satisfies the definition. To more concisely formulate our hypothesis, in the next section, we introduce some concepts from category theory, and then in section 3.3, utilize those concepts to define analogy. The concepts of category theory, such as category or functor, are useful to formulate our hypothesis because those concepts make it possible to concisely summarize the quadruplets of sets $< A, B, R_A, R_B$, or the statement "two functions $f : A \to B, F : R_A \to R_B$ satisfy $F(g) \circ f = f \circ g$".

3.2 Basic concepts of category theory

In this section, we introduce the basic concepts of category theory, to specify our idea of analogy more rigorously. We referred to (Awodey, 2010; Leinster, 2014) to write this section. First, *category* refers to an entity that consists of "objects" and "arrows" such as the ones below.

- objects: $A, B, C, ...$
- arrows: $f, g, h, ...$

Every arrow has a "domain", written as $dom(f)$, and "codomain", written as $cod(f)$. When we write $f : A \to B$, this indicates that $dom(f) = A$ and $cod(f) = B$. *Composition* is the operation that can be applied to arrows, and for $f : A \to B$, $g : B \to C$, the operation is defined as:

$$g \circ f : A \to C$$

This means that arrows f and g have the same object B as $cod(f)$ and $dom(g)$.

The arrow that has the same domain and codomain is called *identity* arrow.

$$1_A : A \to A$$

Every category has to satisfy the following three conditions:

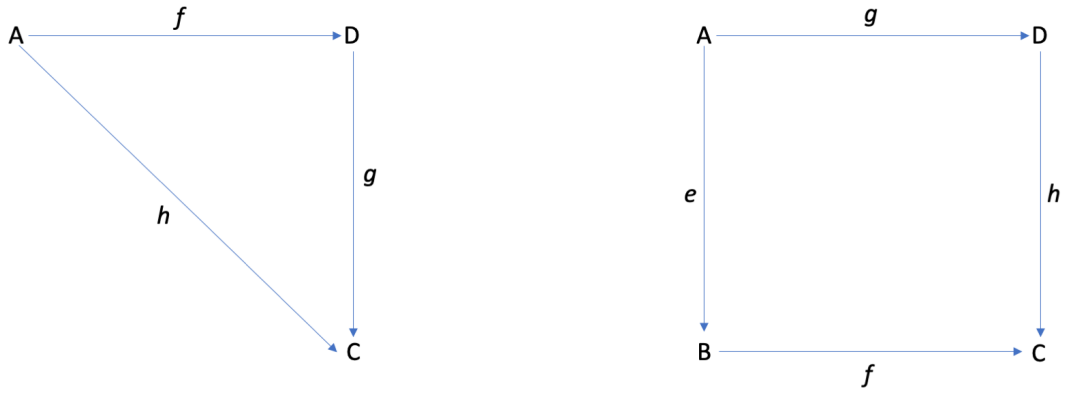


FIGURE 3.1: Examples of commutative diagram

- for every object A, there is an identity arrow $1_A : A \to A$
- for every $f : A \to B, g : B \to C, h : C \to D$, the compositions of the arrows satisfy associative law: $h \circ (g \circ f) = (h \circ g) \circ f$ where
- for every $f : A \to B, 1_A : A \to A, 1_B : B \to B$, there is an identity arrow: $f \circ 1_A = 1_B \circ f$

Here, A, B, C, D are objects. An important concept in category theory is commutative diagram, which represents the compositions between arrows in terms of the directed graph. The commutative diagram is important because it is a useful tool for proving that associativity holds in a category.

Examples of commutative diagram are presented in figure 3.1. Basically, diagram commutes when there are multiple paths on the graph of arrow compositions from an object X to an object Y, the output of compositions on a path is always the same as the output of another path. For example, the right diagram of figure 3.1 commutes because as of two paths from A to C the output of compositions $f \circ e = A \to C = h \circ g$.

The above definition lays the foundation of category and arrows. One example of the category is **Sets**. As the name indicates, **Sets** is the category that has sets $X, Y, ...$ as objects and functions $f : X \to Y, g : YX \to Z, ...$ between sets as arrows. Composition of f and g is defined exactly as composition between functions:

$$\begin{aligned} g \circ f : X &\to Z \\ &\to g(f(x)) \end{aligned}$$

In addition to category, *functor* is also an important concept from category theory. Functor is basically an arrow from a category to another category, that preserves domain and codomain, identity arrow, and composition, of a category. Concretely, given categories $\mathbf{C}, \mathbf{D}$, functor $F : \mathbf{C} \to \mathbf{D}$ consists of two functions:

- A function $\mathbf{C} \mapsto F(\mathbf{C})$
  $$ob(\mathbf{C}) \mapsto ob(\mathbf{D})$$

- A function $\mathbf{f} \mapsto F(\mathbf{f})$ for $A, A' \in \mathbf{C}$

$$\mathbf{C}(A, A') \to \mathbf{D}(F(A), F(A'))$$

Here, ob(**C**) denotes a set of objects in a category, and **C**(A, A') denotes a set arrows in a category. These functions correspond to a function that maps objects to objects, and a function that maps arrows to arrows. Functor also satisfies the following conditions.

- $F(f : A \to B) = F(f) : F(A) \to F(B)$
- $F(1_A) = 1_{F(A)}$
- $F(g \circ f) = F(g) \circ F(f)$

This concludes the introduction of category theory. Next, we apply the concepts from category theory to formulate analogy.

3.3 Our formulation of analogy

In this section, we formulate the computational problem of analogy in terms of the category theoretic notions introduced above. As stated in the section 3.1, we hypothesized that analogy is to define the sets and the functions, under the constraint of commutativity. Category theory sets the stage perfectly to formally specify our hypothesis, with the concepts such as category, arrows and functor.

Using the category theoretic notions above, we can more concisely define analogy as follows, given a set X:

- from X, construct two categories C, D, so that there is a functor $F : C \to D$
- select the categories C, D which have largest number of objects and arrows

The definition concisely captures what was written in the hypothesis, by making tuples A, R_A and B, R_B categories and making a condition "two functions $f : A \to B, F : R_A \to R_B$ satisfy $F(g) \circ f = f \circ g$" an existence of a functor. Let us explain the definition using the "room analogy" example.

3.3.1 Explaining room analogy through the formulation

According to our definition, given a set X, analogy is to construct two categories, so that there is a functor from a category to another category. The functor consists of two functions, one is for objects and the other is for arrows. Let us apply this to room analogy. Here we assume that a vector-space $\mathbb{R}^2$ consists of the coordinates in two dimensions where there are points in figure 3.2. Analogy is to:

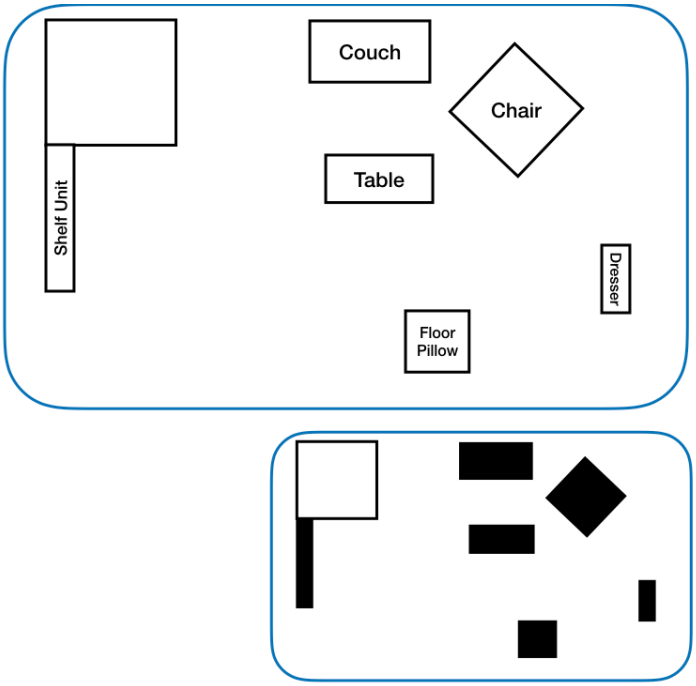


FIGURE 3.2: Same figure as figure 1.1

- from X, construct $roomA, roomB$ so that there is a functor $G : roomA \to roomB$
- then select $roomA, roomB$ which have largest number of elements

Here, $roomA, roomB$ are categories which have points in $\mathbb{R}^2$ as objects, and affine transformations between those points as arrows. Given $\mathbf{y} \in \mathbb{R}^n$,$A \in \mathbb{R}^{n\times n}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^n$ affine transformation refers to the following function f:

$$\mathbf{y} = f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$$

Since we consider the two dimensional case in room analogy, we assume $A \in \mathbb{R}^{2\times 2}$, $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{b} \in \mathbb{R}^2$ from here on. Crucially for this example, affine transformation can describe the basic geometric transformations, such as translation and scaling.

First, we created the set X as in figure 3.3. The figure does not actually contain *all* the points of figure 3.2, but it is simplified to be composed of $48 \in \mathbb{R}$ points. To construct categories, we need to discover the functor, since the functor consists of a constraint to construct the categories, which is supposed to be there from $roomA$ to $roomB$. As is noted in the previous section, functor consists of two functions, one for from object to object, and the other for from arrow to arrow. The problem, then, is to calculate these function from the numerical information given in the figure 3.2 alone. Let's think about the object function first. To do this, we can solve the linear equations derived from the two dimensional affine transformation. Two dimensional affine transformation has six unknown variables, as can be seen below.

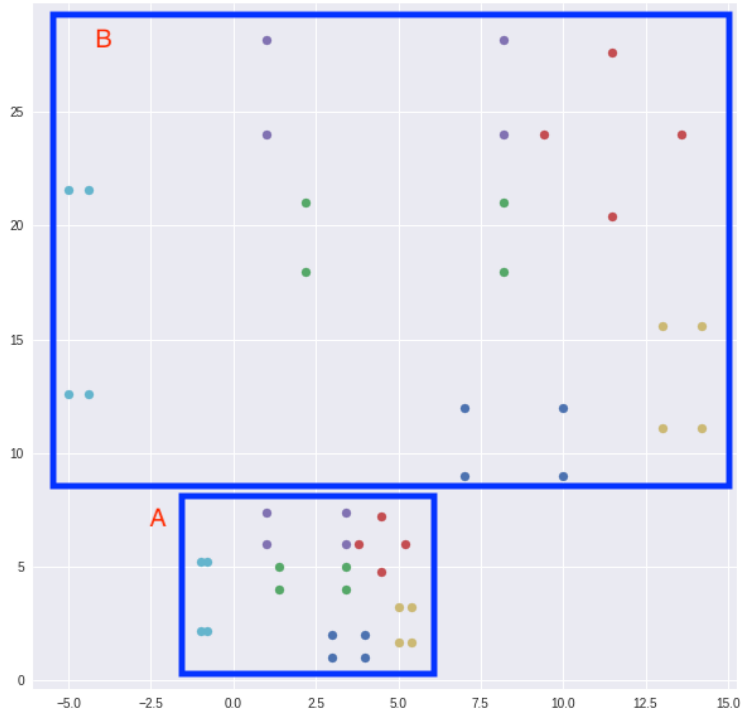


FIGURE 3.3: a set X from the figure 3.2

$$\mathbf{y} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \mathbf{x} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{3.1}$$

Here, $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2$ are the unknown variables. To solve for these variables, we need six equations as well. We can get these equations by using, for example, the three points of dark blue in figure 3.3. The points are $\begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \end{pmatrix}$. By putting each point to the equation 3.1, we get the equations below.

$$\begin{aligned} 3a_{11} + 2a_{12} + b_1 &= 10 \\ 3a_{21} + 2a_{22} + b_2 &= 0 \\ 4a_{11} + 1a_{12} + b_1 &= 15 \\ 4a_{21} + 1a_{22} + b_2 &= -5 \\ 4a_{11} + 2a_{12} + b_1 &= 15 \\ 4a_{21} + 2a_{22} + b_2 &= 0 \end{aligned}$$

By solving for the variables, we get the exact affine transformation that can scale the of points A to the set of points B.

$$f(\mathbf{x}) = \mathbf{y} = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} -2 \\ 6 \end{pmatrix}$$

Thus, we could compute the object function from the figure 3.2. Next, we

want to compute the arrow function. To do this, let us remember the commutativity diagram figure 3.1. Since every functor needs to preserve this commutativity, we can utilize this property to obtain an equation from the constraint $g \circ f = f \circ F(g)$. Here, we can calculate g, for example, as the exactly same way as we calculated f, only changing an used input from three dark blue points on A,B to three dark blue points and three yellow points. From the calculation, we get:

$$g(\mathbf{x}) = \begin{pmatrix} 4 & 0 \\ 0 & 1.5 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 16 \\ 6 \end{pmatrix}$$

By replacing all the functions in $g \circ f = f \circ F(g)$ with affine transformations, we get

$$B(Ax + a) + b = A(Cx + c) + a \tag{3.2}$$

Replacing all the known transformations we get:

$$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} (\begin{pmatrix} 4 & 0 \\ 0 & 1.5 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 16 \\ 6 \end{pmatrix}) + \begin{pmatrix} -2 \\ 6 \end{pmatrix} =$$
$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} (\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} -2 \\ 6 \end{pmatrix}) + \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

From this, we can calculate $c_{11} = 4, c_{12} = 0, c_{21} = 0, c_{22} = 1.5$, and $c_1 = 54, c_2 = 15$. Since F is determined as a function of g, this gives us the arrow function F.

$$F(g) = \begin{pmatrix} 4 & 0 \\ 0 & 1.5 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 54 \\ 15 \end{pmatrix}$$

This gives us the functor G, which consists of the two functions f, F. Thus, it is shown that there is a functor in the constructed categories A, B in figure 3.2.

In this chapter, we have proposed our hypothesis and formulation of analogy, and explained the formulation using a simple example. In the next chapter, we examine the models introduced in chapter 2, in terms of whether the models satisfy the requirements posed in this chapter.

Chapter 4

Do previous models of analogy satisfy the requirements of our definition?

In the previous chapter, we presented our definition of analogy and explained why it might be a candidate as a theory of analogy. In this chapter, we analyze the models of analogy already introduced in Chapter 2, in terms of whether those models satisfy the requirements of our definition and argue that although SME does not satisfy our definition, since it lacks the ability to create it's own category, word2vec can satisfy the important parts, although not all, of our requirements. Based on this analysis, we propose a novel vector operation to solve four-term analogy problems introduced in chapter 2 in word2vec based on our definition, and show that the operation highly improves the performance of analogy in word2vec.

To reiterate, in the previous chapter we proposed the following definition. Given a set X:

- from X, construct two categories C, D, so that there is a functor $F : C \to D$
- select the categories C, D which have largest number of objects and arrows

From the definition, we derive three requirements for the models of analogy to be said to satisfy our definition.

1. being able to construct a functor $F : C \to D$
2. being able to construct categories C, D from X
    (a) being able to utilize a functor as a constraint to construct C, D

First requirement is due to the demand of there being a functor from C to D. Second requirement is obviously derived from the fact that the definition demands the construction of categories C, D. Requirement 2a is due to the fact that, not only a model needs to construct a functor, but it must also be able to utilize it to, in turn, construct the categories. We utilize the requirements to examine two models SME and word2vec.

4.1 Examining SME and word2vec on whether the models satisfy the requirements of our definition

4.1.1 Structure-mapping engine

In this section, we examine structure-mapping engine and word2vec, as to whether the models satisfy the requirements of our definition. To reiterate from Chapter 2, SME is a representative model of analogy first introduced by Falkenhainer, Forbus, and Gentner, 1989 and being developed even now (e.g. Forbus et al., 2017). The model is an implementation of the idea of Gentner, 1983, who states that analogy is to construct the mapping from base domain to target domain, based on the system of relations. To implement the idea, the model takes the structured representations (i.e. directed acyclic graph, where nodes correspond to predicates, as can be seen in (figure 2.3)) as input, and construct the mapping which tries to preserve the higher order relation (predicate of predicate, such as CAUSE in figure 2.3).

The question we ask here is whether the model satisfies our requirements. We argue that the model does not satisfy requirement 2 and 2a, only satisfying 1. SME does satisfy the first requirement, because, first, SME is given directed acyclic graph (DAG) as input, and DAG can be thought of category having nodes as objects and edges as arrows, second, SME can create a mapping from a DAG A to another DAG B, which maps nodes of A as well as edges to nodes and edges of B. Thus, SME satisfies the first requirement. However, the model does not satisfy the requirement 2 and 2a, because it lacks the ability to define the categories on its own from the input. Instead, as pointed out by (Chalmers, French, and Hofstadter, 1992), the model relies on the modelers to supply the inputs already structured as DAG, such as the one in (2.3). Since the second requirement is not satisfied, requirement 2a is not satisfied as well.

4.1.2 word2vec

As it is reviewed in 2, vector-space models refer to the models which learn the distributed (vector-space) representation of words from the corpus, by compressing the cooccurrence probability. Some models of this class, such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington, Socher, and Manning, 2014) are highlighted as a potential model of analogy for its ability to solve a four-term analogy problems introduced in chapter 2, such as Man : Woman :: King : ?, by adding and subtracting word vectors as in 2.6. We argue that the model partially satisfies our requirements, but needs a modification to complete the requirements.

Let us examine the model next. First, given a set X, in this case the set of one-hot vectors, the model can construct categories (categories of vector-space), unlike SME, from the set. In addition, you can specify a functor in the defined categories, in fact, the idea of parallelogram model can be seen

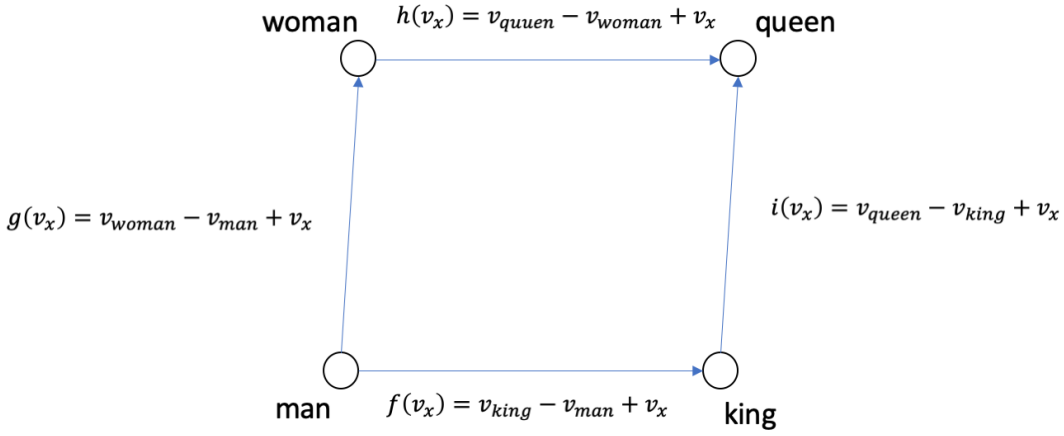


FIGURE 4.1

as specifying a functor. To see this, think of the four translation functions $f(v_x) = v_{king} - v_{man} + v_x, g(v_x) = v_{woman} - v_{man} + v_x, h(v_x) = v_{queen} - v_{woman} + v_x, i(v_x) = v_{queen} - v_{king} + v_x$, also depicted in figure 4.1. The functions constitute a parallelogram relationship among four words, "man", "woman", "king", "queen", as specified by parallelogram model. These functions are commutative, since $i \circ f = v_{queen} - v_{man} + v_x = h \circ g$, and thus constitute a functor. Therefore, the fact that the model can solve analogy problems utilizing this kind of relationship shows that the model can at least approximately captures the functor in the sense of figure 4.1. From this, we can conclude that word2vec satisfies the requirements 1 and 2.

However, the model does not utilize the functor itself to construct and select the vector-space. Rather, the model creates the vector-space through approximating the cooccurrence statistics, and then that created vector-space captures the functor relations between word vectors. Thus, although word2vec satisfies the requirements 1 and 2, in that the model can create the representation on which functor can be defined, it does not satisfy the requirement 2a in that the model does not utilize the constraint posed by functor in itself.

In this section, we have analyzed the two models of analogy in terms of whether these models satisfy our requirement of analogy. We have argued that both SME and word2vec does not dully qualify as a model of analogy on our requirement, however, word2vec does satisfy conditions (1), (2). According to the analysis of word2vec above, we can consider the parallelogram relationship in chapter 2 as an example of functor. Based on this observation, in the next section we devise a new analogy operator for word2vec, which better captures the parallelogram relationship.

4.2 Deriving the analogical operator from the definition

In the last section, we have suggested that parallelogram relationship can be seen as an example of functor. In this section, we derive a new analogical inference operator based on this observation, and test the performance of the

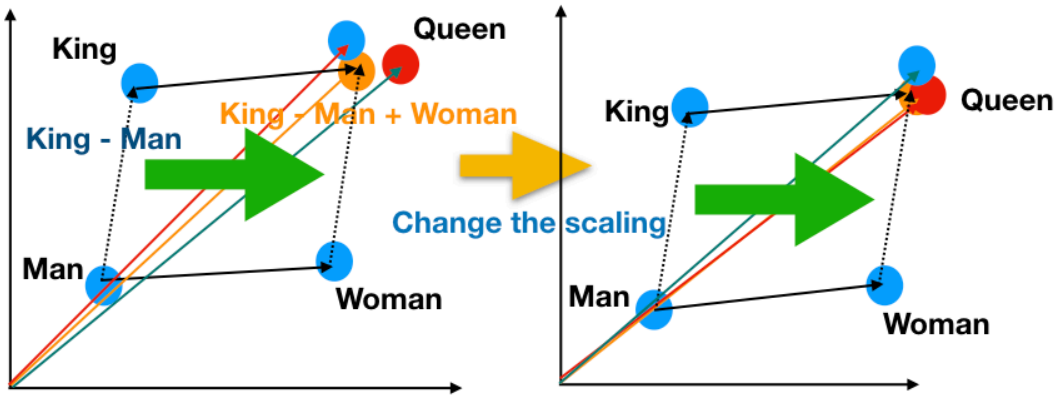


FIGURE 4.2: Our method changes the scaling of given word vectors, so that it preserves the parallelogram relation and at the same time separate the correct word vector from noise vectors.

operator. We find that our operator highly improves the performance of analogy in the model compared to the previously proposed operators. We first review the previous methods for performing analogical inference, and then derive our own method. We presented this result here (Kato and Hidaka, 2018).

4.2.1 Background and Motivation

As it is noted in section 2.2.5, vector-space models utilize an analogical inference operator, such as 2.6, or 2.7, to solve four-term analogy problems. Here, both methods are presented again:

$$f(v_a, v_b, v_c) = \underset{v_d \notin \{v_a, v_b, v_c\}}{\arg\max} \left(\cos(v_c - v_a + v_b, v_d)\right) \tag{4.1}$$

$$g(v_a, v_b, v_c) = \underset{v_d \notin \{v_a, v_b, v_c\}}{\arg\max} \frac{\cos(v_b, v_d)\cos(v_c, v_d)}{\cos(v_a, v_d)} \tag{4.2}$$

Equation 4.1 is due to (Mikolov et al., 2013a), while equation 4.2 is to (Levy and Goldberg, 2014). Both methods are based only on the translation of word vectors in the given space. However, our definition of analogy allows not only the use of translation, but also the use of any transformation applied to word vectors to perform analogy. Based on this idea, we generalize the equation 4.1 so that analogy operator also includes scaling transformation to the given word vectors. We estimate scaling transformation via supervised-learning on a given set of words in an analogical relationship.

4.2.2 Method

Our method utilizes the supervised learning to scale a given words in analogy problems so that analogical inference is more accurate (figure 4.2). We

define the analogical inference operator as follows:

$$h_{M_1,M_2}(v_a, v_b, v_c) = \underset{v_d \notin \{v_a, v_b, v_c\}}{\arg\max} \left(\cos(M_1 v_c - M_1 v_a + M_2 v_b, M_2 v_d)\right) \tag{4.3}$$

Here M_i is a diagonal matrix, and each diagonal value of M_i is composed of some weight w_i. The formulation can be viewed as a generalized version of method 4.1, since if we take $M_1 = M_2 = I$, where I is identity matrix, 4.1 corresponds exactly to 4.1. The important part of our method is the selection process of w_i in M_i, since arbitrarily selecting w_i doesn't make analogical inference more accurate.

Let us explain the selection process of w_i in M_i. First, the elements of a large number of word vectors learned through skip-gram take values near zero, and fewer word vectors take large values in the same dimension. As only one word vector is the "correct" answer in any analogy problem, the vast majority of other words is a "noise" [1]. The empirical distribution of this "noise" word vectors in a particular dimension follows an exponential distribution. An example of the values of the first dimension of the noise word vectors are shown in the figure 4.3. Therefore, considering there are fewer word vectors which take larger absolute values, it is easier to separate out the correct word vector from other noise vectors. From this observation, we gain a rule of thumb that we should choose some dimensions on which many word vectors in the test set have larger absolute values.

Secondly, we choose some subspace in which the words in the test set form the "analogical relationship" which the analogy inference operator 4.3 will identify as the answer. For D dimensional vector space of N words, let $\mathbf{V}_0 \in \mathbb{R}^{K \times D}$ be a matrix of K word vectors, in which each row has a word vector of some class from the google test set. For example a class of word vectors which (male matrix), and let $\mathbf{V}_1 \in \mathbb{R}^{K \times D}$ be a matrix paired with $\mathbf{V}_0$, in which i^{th} row has a vector corresponding i^{th} row in $\mathbf{V}_0$, such as woman and queen ("female" matrix). Then, for the model applying the function 4.3, the dimension j which has smaller error ϵ_j defined as $\epsilon_j = \|\mathbf{V}_{0,j} - \mathbf{V}_{1,j} + \mathbf{1}_K c^\top\|_2$, is more preferable, where $c \in \mathbb{R}^D$ is some translation vector minimizing ϵ_j with respect to c, and $\mathbf{1}_K \in \mathbb{R}^K$ is the vector with its all elements being 1. Taking a subspace of dimension with $\epsilon_j = 0$, the analogy inference of equation 4.1 exactly identifies the correct answer for the given triplet of word vectors in the given analogy problem.

Summarizing two general preferences to have a better vector-space:

1. choose dimensions in which words in test set have larger absolute values
2. choose the dimension i in which words in test set have lower ϵ_i

[1] if we solve an analogy problem utilizing the vector-space of three million word vectors, 299996 word vectors are noise, excluding the given three word vectors in the problem, and the correct word vector.

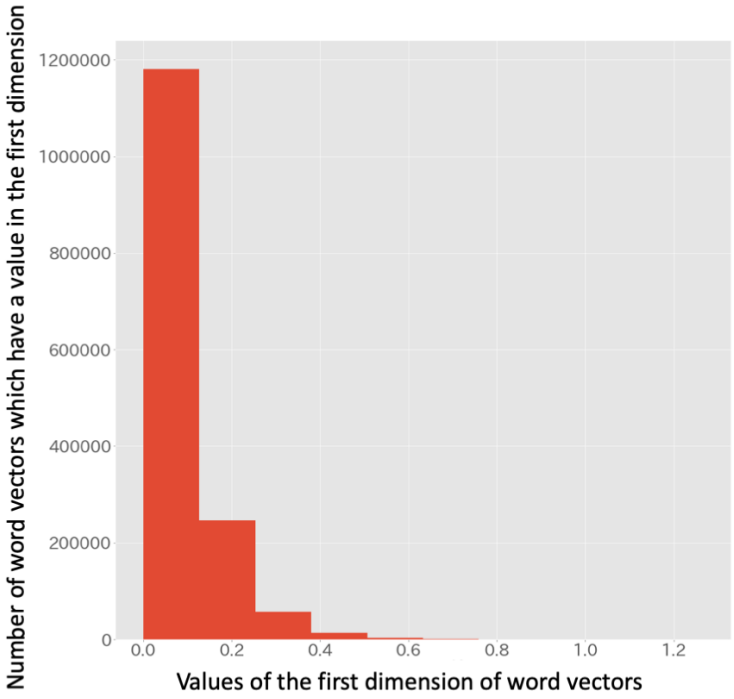


FIGURE 4.3: The empirical distribution of the first dimension of "noise" word vectors.

Considering the two conditions, as a heuristics, we devised a weight for dimension i below:

$$w_i = e^{\max(|\mathbf{V}_{0,i}|)+\max(|\mathbf{V}_{1.i}|)-\epsilon} \tag{4.4}$$

In 4.4, $\max(|\mathbf{V}_{0,i}|) + \max(|\mathbf{V}_{1.i}|)$ part reflects the preference 1 by taking the maximum of an absolute value of dimensions for each word, and subtracting ϵ reflects the preference 2. In the following experiment, we used the pre-trained word vectors Mikolov et al., 2013d which contain three million of words with each word having 300 dimensions. As analogy set, we utilized widely used google test set, which contains 19544 pairs of analogy questions (8,869 semantic and 10,675 syntactic questions). We calculated weights by 4.4 and applied weights to 300 dimensions of three million word vectors, then obtained the model answer by 4.3 with weighted vectors.

4.2.3 Result

The analogy performance of three methods, 4.2, 4.1, 4.3, is shown on figure 4.2. Our method outperforms other two methods on all the word categories, except for capital-world. Specifically, the accuracy of "currency", "gram1-adjective-to-adverb", "gram2-opposite" increased 20-30%, which have gotten lowest accuracy on Mikolov method.

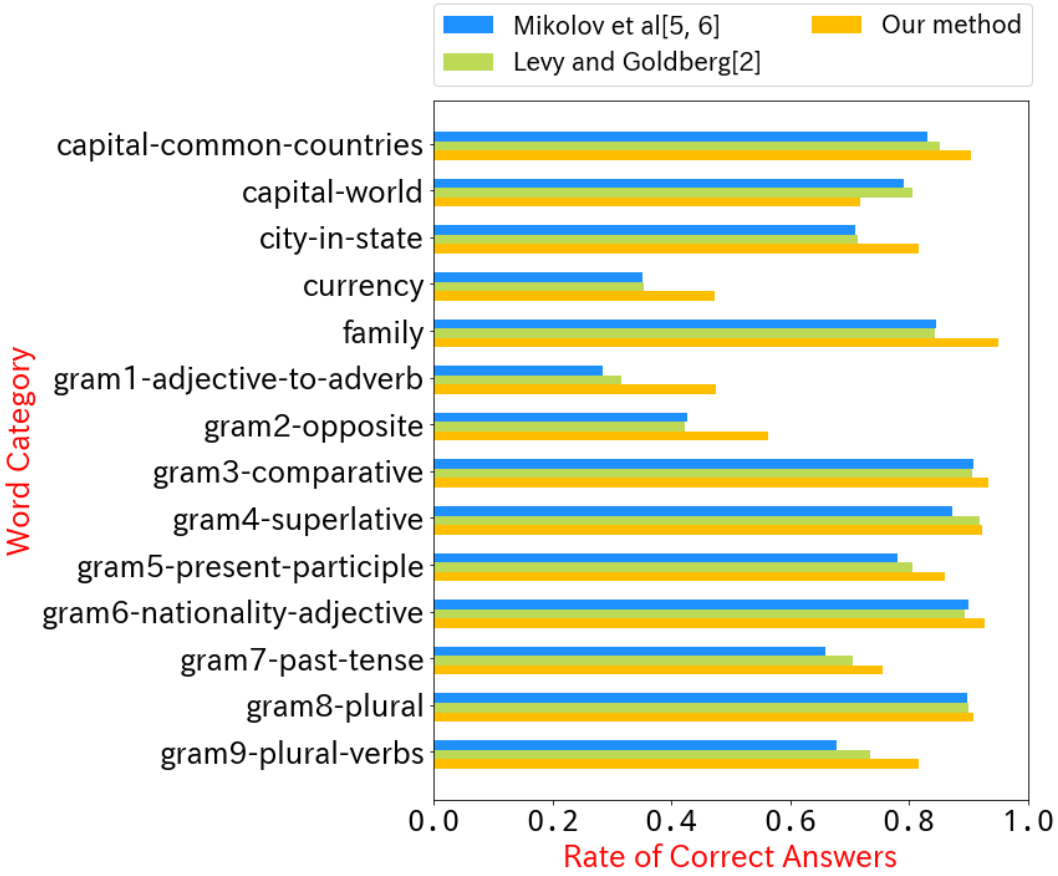


FIGURE 4.4: The result comparing the three methods, 4.2, 4.1, 4.3 with accuracy on google testset.

4.2.4 Discussion

Using our method, we have gained substantive increase in accuracy on google testset compared to previously proposed methods. The result may indicate that our definition of analogy as constructing a functor is on a right track, since it verifies the prediction of our definition that relations can be better represented as arrows, not some specific function such as translation.

Chapter 5

Discussion and Conclusions

In this paper, we have reviewed the previous researches on the theorization and modeling of analogy (Chapter 2), proposed our hypothesis and definition of analogy (Chapter 3), and examined the previous models whether they satisfy the requirements of the definition (Chapter 4).

The critical question we have tried to tackle in this paper is in making an analogy, "where does the relation come from?". The similar critique towards SME is raised by (Chalmers, French, and Hofstadter, 1992), that the previous models of analogy has lacked the ability to create the representation utilized in analogy in itself, allowing modelers to arbitrarily create the representation leading to the trivialization of solving analogy. To address the question, we have pointed out that the inability of the models to create the representation is due to the underspecification of the structure-mapping theory, and proposed the new definition of analogy. Through this definition, we have tried to capture what was lacking in SMT, which is how the representation used in analogy should be created, without assuming nothing other than a set. Our definition gave the requirements to the process of analogy as creating the categories, which can be thought of as a set of objects with relations, from a given set, so that there is a functor from a created category to another created category. We have tested this idea on a simple example, room analogy, and demonstrated that we can construct the categories, as required by the definition. After this, we have analyzed SME and word2vec, and suggested that although SME does not satisfy our requirements of being the model of analogy, word2vec can partially satisfy the requirements. Based on this observation, we tested the prediction of our definition, analogy can be better captured through the use of an arrow as a relation, by generalizing the analogical operator of (Mikolov et al., 2013a). Our test showed the preliminary success of the operator over previous ones.

As we have pointed out, word2vec satisfies the two requirements of our definition. So, what can be done to create the model that completely satisfies our theory? Our requirements demand that the model should be able to create the categories from some set with the constraint of there being a functor. The researches on knowledge graph embedding (Wang et al., 2017) show some of the possible directions on this line. Knowledge graph embedding refers to a technique which learns the vector-space representation of the graph structures, which have entities as nodes and relations between entities as edges. For example, (Liu, Wu, and Yang, 2017) proposed a model

for knowledge graph embedding called *ANALOGY*, which learns the embedding under the constraint of commutativity of edges. These models do not satisfy our requirements by themselves, because it requires the input structured as knowledge graph, which is same as what SME does. However, the way the models build vector representation aligns with what we propose in our formulation, because it utilizes the constraint such as commutativity.

Overall, our proposed definition has some advantages over SMT, such as the specification of how to build the representation, and has shown some promising results on the preliminary testings presented in this paper. Thus, we believe our definition is worth further testing. To more comprehensively test our definition, we need to build the model which completely satisfies our proposed requirements.

Bibliography

Awodey, Steve (2010). *Category theory*. Oxford University Press.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski (2014). "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors". en. In: Association for Computational Linguistics, pp. 238–247.

Bruni, E., N. K. Tran, and M. Baroni (2014). "Multimodal Distributional Semantics". In: *Journal of Artificial Intelligence Research* 49, pp. 1–47.

Chalmers, David J., Robert M. French, and Douglas R. Hofstadter (1992). "High-level perception, representation, and analogy: A critique of artificial intelligence methodology". In: *Journal of Experimental Theoretical Artificial Intelligence* 4.3, pp. 185–211.

Chen, Dawn, Joshua C Peterson, and Thomas L Griffiths (2016). "Evaluating vector-space models of analogy". en. In:

DeLoache, Judy S (1989). "Young children's understanding of the correspondence between a scale model and a larger space". In: *Cognitive Development* 4.2, pp. 121–139.

Encyclopaedia Britannica, The Editors of (Dec. 2018). *Rutherford atomic model*. URL: https://www.britannica.com/science/Rutherford-atomic-model.

Falkenhainer, Brian, Kenneth D. Forbus, and Dedre Gentner (1989). "The structure-mapping engine: Algorithm and examples". In: *Artificial Intelligence* 41.1, pp. 1–63.

Finkelstein, Lev et al. (2002). "Placing search in context: The concept revisited". In: *ACM Transactions on information systems* 20.1, pp. 116–131.

Firth, J. R (1957). "A synopsis of linguistic theory". In: *Studies in linguistic analysis*. Oxford: Blackwell.

Forbus, Kenneth D. et al. (2017). "Extending SME to Handle Large-Scale Cognitive Modeling". In: *Cognitive Science* 41.5, pp. 1152–1201.

Gentner, Dedre (1983). "Structure-Mapping: A Theoretical Framework for Analogy*". In: *Cognitive Science* 7.2, pp. 155–170. ISSN: 03640213.

— (July 2010). "Bootstrapping the Mind: Analogical Processes and Symbol Systems". en. In: *Cognitive Science* 34.5, pp. 752–775.

Gentner, Dedre and Kenneth D. Forbus (May 2011). "Computational models of analogy: Computational models of analogy". en. In: *Wiley Interdisciplinary Reviews: Cognitive Science* 2.3, pp. 266–276.

Gentner, Dedre and Linsey A. Smith (2013). *Analogical Learning and Reasoning*. Oxford University Press.

Gentner, Dedre et al. (2001). "Metaphor is like analogy". In: *The analogical mind: Perspectives from cognitive science*, pp. 199–253.

Henley, Nancy M (1969). "A psychological study of the semantics of animal terms". In: *Journal of Verbal Learning and Verbal Behavior* 8.2, pp. 176–184.

Hill, Felix, Roi Reichart, and Anna Korhonen (Aug. 2014). "SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation". en. In: *arXiv:1408.3456 [cs]*. (Visited on 01/11/2019).

Holyoak, Keith J, Dedre Gentner, and Boicho N Kokinov (2001). "Introduction: The Place of Analogy in Cognition". In: p. 21.

Holyoak, Keith J, Keith James Holyoak, and Paul Thagard (1995). *Mental leaps: Analogy in creative thought*. MIT press.

Holyoak, Keith J. and Paul Thagard (1989). "Analogical Mapping by Constraint Satisfaction". In: *Cognitive Science* 13.3, pp. 295–355.

Hummel, John E. and Keith J. Holyoak (2003). "A symbolic-connectionist theory of relational inference and generalization." In: *Psychological Review* 110.2, pp. 220–264.

Kato, Tatsuhiko and Shohei Hidaka (2018). "Improving Analogical Inference Using Vector Operations with Adaptive Weights". en. In: *The Proceedings of the 28 th Annual Conference of the Japanese Neural Network Society*, p. 124.

Leinster, Tom (2014). *Basic category theory*. Vol. 143. Cambridge University Press.

Levy, Omer and Yoav Goldberg (Apr. 2014). "Linguistic Regularities in Sparse and Explicit Word Representations". In: *Proceedings of the eighteenth conference on computational natural language learning*, pp. 171–180.

Linzen, Tal (2016). "Issues in Evaluating Semantic Spaces Using Word Analogies". In: *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, pp. 13–18.

Liu, Hanxiao, Yuexin Wu, and Yiming Yang (2017). "Analogical Inference for Multi-Relational Embeddings". In: *arXiv:1705.02426 [cs]*.

Mikolov, T M et al. (2013a). "Distributed Representations of Words and Phrases and their Compositionality". In: *NIPS*, pp. 1–9.

Mikolov, T M et al. (2013b). "Efficient Estimation of Word Representations in Vector Space ". In: *ICLR Workshop Papers*, pp. 1–12.

— (2013c). *Google Analogy Testset*.

— (2013d). *Pre-trained word2vec vectors*.

Penn, Derek C., Keith J. Holyoak, and Daniel J. Povinelli (2008). "Darwin's mistake: Explaining the discontinuity between human and nonhuman minds". In: *Behavioral and Brain Sciences* 31.02.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Rumelhart, David E and A Abrahamsen (1973). "A Model for Analogical Reasoning". In: *Cognitive Psychology* 5, pp. 1–28.

Speer, Robert and Catherine Havasi (2012). "Representing General Relational Knowledge in ConceptNet 5." In: *LREC*, pp. 3679–3686.

Turney, Peter D and Patrick Pantel (2010). "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37, pp. 141–188.

Wang, Quan et al. (2017). "Knowledge Graph Embedding: A Survey of Approaches and Applications". In: *IEEE Transactions on Knowledge and Data Engineering* 29.12, pp. 2724–2743.

Young, Tom et al. (2017). "Recent Trends in Deep Learning Based Natural Language Processing". In: *arXiv:1708.02709 [cs]*.