Digitalization of Mathematical Textbooks and Evaluation of the Technology

Hideto IKEDA
Department of Computer Science
Ritsumeikan University
Japan
hikeda@is.ritsumei.ac.jp

Naoya Idota
Department of Computer Science
Ritsumeikan University
Japan
n.idota0127@gmail.com

Ngyen Thang Hung
Department of Computer Science
Ritsumeikan University
Japan

Ze Zhong Li
Department of Computer Science
Ritsumeikan University
Japan
lizezhonglaile@163.com

Abstract— This paper proposes a language description model to describe sentence structures by a sequence of phrases. For digitalization of mathematical documents, it is important to digitalize natural sentence parts, not just mathematical formulas. The digitalization should be language independent. By using para-phrase patterns, it is possible to implement it. The remain issue is to construct phrase dictionary. This paper also discusses how to extract para-phrase patterns from parallel corpuses. For the evaluation of this approach, digitalization of a textbook of discreet mathematics is done by this approach.

Keywords- Digitization of Mathematical textbook, Para-Phrase Model of sentence, eLearning, Natural Language Processing, Machine translation

I. INTRODUCTION

Digitalization of mathematical documents is one of hot topics in eLearning, because mathematics is the theoretical base of every science. Historically digitalization of mathematical documents has been implemented by step by step. First step uses just scan documents as image data and makes digital textbooks as eLearning contents. Learner can read textbooks, but cannot search relevant pages by keywords. The second step uses general OCR and enables to search by keywords, but mathematical formulas were expressed as image data. The 3rd step using mathematical formula recognition software e.g., Infty[4], enables to express mathematical formula by script codes, e.g., MATH-ML and Tex script. It makes deeper recognition of mathematical documents. But natural sentence parts stay as just character strings. The authors think a much deeper recognition of natural sentence parts is the 4th step.

This paper proposes the phrase grammar to express natural sentences in mathematical books by a sequence of phrases. This technology enables us to recognize sentences with mathematically special meaning, e.g., definition, relation, assumption and implication and translate them into other languages.

II. PHRASE ALIGNMENT OF SENMTENCE IN JAPANESE AND ENGLISH

III.

In order to define the phrase grammar, we shall see conventional phrases of a sentence in Japanese and English. As an example of sentence, we use the following aligned sentence;

"*Two vertices, i.e. faces of the original drawing, are joined by an edge in G if they are neighboring faces and the endpoints of their common edge have labels 1 and 2.*"

and

"*2 つの頂点、すなわち、もとの三角形分割の 2 つの領域は、互いに隣接しており,その共有する辺の端点に 1 と 2 が割り当てられているときに,G の辺で結ぶことにする。*".

If we keep all pair of aligned sentences in the database, we can establish language dependent coding for natural sentences appeared in a target document. However, how many sentences do we have to keep in the database? It may be impossible in the current computer. If we focus to word alignment, what will happen? The conventional word alignment can be shown as follows;

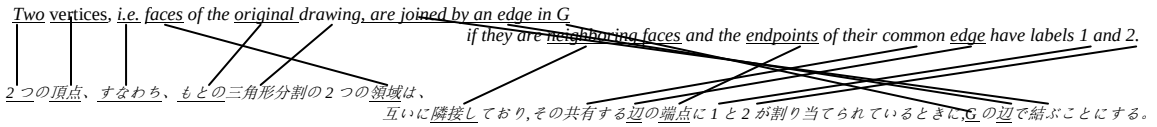


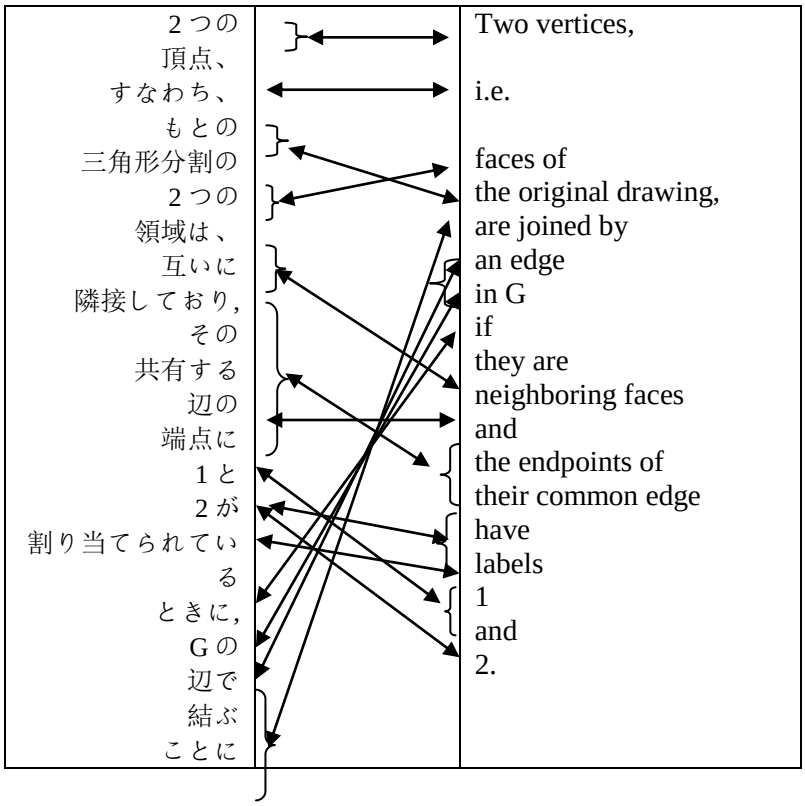
Figure 1. Word alignment of translated sen

As you can see Figure 1, there is not perfect alignment of words, because between two different languages, there are some words that are appeared in just one language. How about phrase alignment?

By a conventional lowest phrase structure of the two sentences are shown in Figure 2. In the phrases, we have the perfect alignment if we combine some phrases into one phrase and making correspondences. But if we will construct pairs of phrase, some non-appropriate alignments, e.g., ("*もとの三角形分割*", "*the original drawing*"), ("*互いに隣接しており*", " *they are neighboring faces and*"), ("*割り当てられている*", "*have Labels*"), and ("結ぶことにする。", "are joined by"). These pairs are valid in only in these sentences and can not use these pairs for translation of other sentences.

This example shows us that we have to define another type of phrase for perfect and effective phrasing. We shall decompose the sentences into higher level phrases in Figure 3.

As you can see Figure 1, there is not perfect alignment of words, because between two different languages, there are some words that are appeared in just one language. How about phrase alignment?



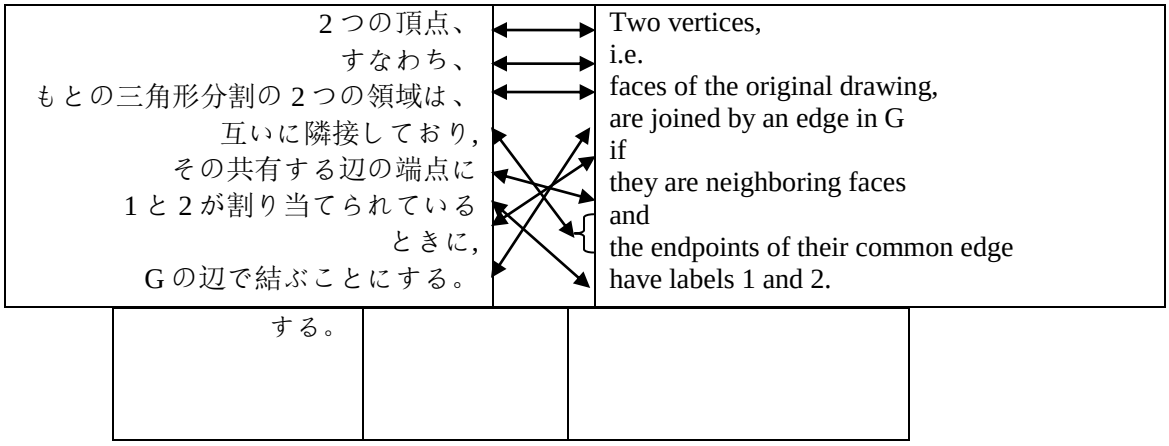


Figure 2 Phrase alignment of Japanese and English sentence

Figure 3 Higher-level Phrase alignments of Japanese and English Sentences

By a conventional lowest phrase structure of the two sentences are shown in Figure 2. In the phrases, we have the perfect alignment if we combine some phrases into one phrase and making correspondences. But if we will construct pairs of phrase, some non-appropriate alignments, e.g., ("*もとの三角形分割*", "*the original drawing*"), ("*互いに隣接しており*", " *they are neighboring faces and*"), ("*割り当てられている*", "*have Labels*"), and ("結ぶことにする。", "are joined by"). These pairs are valid in only in these sentences and can not use these pairs for translation of other sentences.

As translation pairs of phrases, these are acceptable, but we cannot reduce the size of phrase pair database by this approach. If we wish to do it, we need much higher abstraction of phrases. One of solution for the abstraction is to make patterns of phrase. This is common approach for language education. The pattern is described by the following functional forms;

S0=Gの辺で結ぶことにする。	S0=are joined by an edge in G.
S0=@v連体形:ことにする。([P1]) P1=_の辺で結ぶ([N2]) N2=G	S0=_.([P1]) P1= are joined by an edge in _([N2]) N2=G

Figure 4. Example of Phrase Pattern

In Figure 4, underline (_) in a function name is to show a position where a corresponding parameter value is embedded. Although the expression of @v連体形: is also embedded position of parameter

value, the value should be replaced by conjugated form of the parameter value. This kind expression is appeared in only languages with verb conjugation, like Japanese. By this approach to making patterns of concrete phrases, we can reduce the size of the phrase database. The whole phrase structure of the above example is in Figure 5. An important thing of this approach is not just making phrase patterns, but also each phrase is corresponded to a phrase of another language. This type phrase patterns is referred to a para-phrase pattern, or PPP.

N1= 2 つの頂点 N2=_、すなわち、([N1],[N3]) N3=もとの三角形分割の 2 つ領域 P4=_が_の辺で結ぶ([N3],[N6]) N5=G N6=その共有する辺の端点 P7=互いに隣接する P8=_と_が割り当てられている([N9],[N10]) N9=1 N10=2 S11=_は、@v 連用タ形:ており、@v 連体形:ときに、@v 連体形:ことにする。 ([N2],[P7],[P8],[P4])	N1=Two vertices N2=_, i.e. _([N1],[N3]) N3=faces of the original drawing, P4=_ are joined by an edge in _ ([N3],[N5]) N5=G N6=the endpoints of their common edge P7= they are neighboring faces P8=have labels _ and _.([N9],[N10]) N9=1 N10=2 S11=_, _ if _ and _.([N2],[P4],[P7],[P8])

Figure 5. Para Phrase Pattern for Example Sentences

In Figure 5, the last phrase function is the final sentence-level phrase. In conventional phrasing approach, the sentence-level phrases (so-called sentence patterns) have not been recognized. But the final sentence-level para-phrases include the most important know-how of professional translators. Almost concrete noun phrases and basic verb phrases are organized in the dictionary. There is, however, no perfect sentence pattern dictionary.

III CONSTRAINTS OF PHRASE EMDEDDING

Each phrase in PPP has a type. Types of PPP phrase are just three that is, noun (N), predicate (P) and sentence (S). Although we can avoid some of neither illegal phrases nor sentences by using these types, it is not enough to avoid for making incorrect sentences. It is necessary to establish the set of rules to control the correctness of sentences. In conventional grammars, the rules are described by using part-of-speech (POS). But by a POS-based approach, we cannot avoid exceptions. Another approach is to assign a concept code to each parameter of each function. Typical examples of concept code are person-name, organization-name, food, place and so on. This concept code approach for embedding control is better than POS control. It is, however, not perfect also. We adopt concrete-phrase (CP) approach in which each combination of phrases is controlled to be possible to embed or not.

IV PHRASE GRAMMAR AND PARALELL PHRASES

Now we shall define Phrase Grammar of the target of the paper on the base of above considerations.

Let Σ be the character set and Σ^* is the set of all finite strings of elements of Σ. A sentence σ in Σ is a string of Σ^*, $\sigma \in \Sigma^*$. Let S^* is the set of correct sentences, that is $S^* \subset \Sigma^*$. The set S^* can be assumed a priori. In general, a grammar G is a function the set of rules to judge a string is a correct sentence or not. Let L(G) be the set of all strings does not violate any rule. A perfect grammar is a grammar having $L(G) = S^*$. Although many linguists have been trying to find simple description of the rule set to establish a perfect grammar, they have never succeeded.

Phrase grammar is a 3-tuple G(V, Π, R) satisfying the following conditions;

(1) V is a subset of strings in Σ^*,

(2) Π is a subset of strings in $(V \cup \{\_\})^*$,

(3) R is a function of $\Pi \times \Pi \times N \rightarrow \{0,1\}$.

An element of Π is called a phrase pattern and R is the set of all embedding rules.

For two or more phrase grammars $G_1, G_2, \ldots, G_n$, a parallel phrase grammar G(V, Π, R) is defined as follows;

(1) $V \subset V_1 \times V_2 \times \ldots \times V_n$

(2) $\Pi \subset \Pi_1 \times \Pi_2 \times \ldots \times \Pi_n$

(3) $R = \cup R_i$.

V.NUMER OF PHRASE PATTERNS

By the definition of phrase patterns, we can estimate the number of phrase patterns. Basically, phrase patterns can be classified into the following 13 groups;

- Basic Verb Phrases (Ex：P=*catch up _ ([N:object]))
- Basic Adjective Phrases (Ex：S=_ *@be-subj:beautiful ([N:a-object]))
- Modality Phrase (Ex：S=will _([P]))
- Adjective Modifier Phrase(Ex：N=*beautiful _([N]))
- Adverb Modifier Phrase (Ex：P=quickly*([P]))
- Sentence Modifier Phrase (Ex：S=For example, _([S]))
- Noun Phrase (Ex：N=_ of _([N],[N]))
- Conjunction Phrase (Ex：S=_ and _([S],[S]))
- Adverbial Modifier of Functional words (Ex：P=had better to _ ([P]))
- Phrase without main word (Ex：S=_ _([N:agent],[P]))
- Tense and Aspect Phrase (Ex：P=was @pp:([P]), ：P=have been @ing:([P]))
- Sentence Phrase (Ex：S=The _ enjoys popularity among _ because it is _. ([N],[N],[P]))
- Definite Value Phrase (Ex: N=discreet mathematics)

Among these phrase groups, noun phrases, we have more than several million phrases of basic verb phrases, and sentence phrase. We shall show why there are so many phrase pairs in there three groups.

Call _ a mistake	__を誤りとする
Call _ a fictitious story	__を作り話という
Call _ for more detail	__詳細を知りたくて電話する
Call _ hard name	_悪態をつく
Call _ into action	_に行動するように求める
Call _ into being	_を生み出す

Figure 6 Translation of Verb “Call”

Some English verbs, especially non-saturated words (make, take, let, do, have ,…) has many translations and cannot translate without thinking objects. In figure 6, “call” is translated into “とする”, ”という”, ”電話する”, ”つく”, ”求める”, ”出す”. As the standard translation of (“call”, “を呼ぶ”), these above “call” are treated as (“Call _ a mistake”, “__を誤りとする”), …, (“Call _ into being”,” _を生み出す”).

Complex Noun phrases have also such kinds of phenomena. For example, we have (“機能”, “function”) in general domain, but (“関数”, “function”) in mathematics. Almost complex nouns cannot be translated by just using translation of individual noun. Almost technical terms are complex nouns and cannot be translation by just using translation of individual noun.

Since the embedding rule is a 0-1 matrix of phrase pairs, the size of rules have tera-emements. If this size is not very difficult to be managed by the current computers. The most important, but difficult issue is how to collect these para-phrase-patterns.

VI. CONSTRUCTION OF PPP DICTIONARY IN DISCREET MATHEMATICS

We shall show the construction approach of PPP dictionary in a field of discreet mathematics.

もっぱら有限集合を頂点集合とするグラフ	We almost always consider graphs with finite vertex sets.

<table>
<tr><td>を扱う。</td><td></td></tr>
<tr><td><Step-1> Analysis by Dependency Parser Cabocha
* 0 3D 0/0 0.54634510
もっぱら もっぱら 副詞-一般
* 1 3D 1/2 1.67090004
有限 有限 名詞-一般
集合 集合 名詞-サ変接続
を ヲ を 助詞-格助詞-一般
* 2 3D 1/2 2.76115702
頂点 頂点 名詞-一般
集合 集合 名詞-サ変接続
と と 助詞-格助詞-一般
* 3 4D 0/0 1.16913508
する する 動詞-自立 サ変・スル 基本形
* 4 5D 0/1 0.00000000
グラフ グラフ 名詞-一般
を を 助詞-格助詞-一般
* 5 -1O 0/0 0.00000000
扱う 扱う 動詞-自立 五段・ワ行促音便 基本形
。 。 。 記号-句点
EOS</td><td><Step-1> Analysis by Stanford Parser
(ROOT
(S
(NP (PRP we))
(VP
(ADVP (RB almost) (RB always))
(VBP consider)
(NP (NNS graphs))
(PP (IN with)
(NP (JJ finite) (NN vertex) (NNS sets)))) (. .)))

<Step-2> Functionalization
S0=_ _.([NP1],[VP2])
NP1=we
VP2=_ _([ADVP3],[VBP4])
ADVP3=_ _([RB5],[RB5])
RB5= almost
RB6= always
VB9=consider _ _([NP10],[PP11])
NP10= _NNS12
PP11=graphs with _([NP13])
NP13=_ _ _([JJ14],[NN15],[NNS16])
JJ14=finite
NN15= vertex
NNS16= sets</td></tr>
<tr><td><Step-2> Functionalization
N1=有限集合
N2=頂点集合
P3=_を_とする([N1],[N2])
N4=@v基本形:グラフ([P3])
P5=_を扱う。([N4])</td><td><Step-3> Delete functional words
S0=_ almost always consider _.([N1],[N2])
N1=we
N2=graphs with finite vertex sets</td></tr>
<tr><td><Step-3> Delete functional words
N4=有限集合を頂点集合とするグラフ
P5=_を扱う。([N4])</td><td></td></tr>
<tr><td><Step-4> Alignment
P5=_を扱う。([N4])
N4=有限集合を頂点集合とするグラフ</td><td><Step-4> Alignment
S0=we almost always consider _.([N2])
N2=graphs with finite vertex sets</td></tr>
</table>

Figure 7 Extraction of PPP from Parallel Corpus

VII. ASSIGNMENT OF SPECIAL MEANING IN MATRHEMATICS TO PHRASES

For each extracted para-phrase, some special meaning in mathematics is assigned as attributes. For above example, we attach attribute “topic” to the following sentence phrase. The set of all attributes is shown in Figure 7.

andList	ordering
apply	pair
assumption	principleOfProof
andList	proofByCase
apply	proofByContradiction
assumption	topic

calculation	quotation
conclusion	ratio
condition	reference
conditionOfAboveSentence	result
consider	similar
equivalence	substitution
fact	syllogismcalculation
G_definition	conclusion
imply	condition
induction	conditionOfAboveSentence
L_definition	consider
not	equivalence
notice	
orList	

Figure 8 Attribute List of Sentence–level Phrases

For a noun phrase, we can attach the following attribute in Figure 9.

agent	inequality
algorithm	infinite
andList	L-definition
a-object	limit
basis	max
calculation	min
chart	mod
commonMultiple	modifier
condition	number
conparison	numberWithCondition
decrease	orList
definition	ordering
division	production
equation	proof
example	quotation
exponent	quotient
expression	rationalNumber
expressionNumber	recurrence
expressionWithCon	referenceNo
dition	remainder
factorG_definition	result
imply	sequence
induction	series
L_definition	set
not	substitution
notice	supplementaryConditio
orList	n
factorization	nounWithSupplementa
false	ryCondition
figure	symbol
finite	symbolWithCondition
formula	theorem
function	true
G-definition	value
group	
increase	

VIII. CONCLUTIONS AND DISCUSSION

This paper proposes the sentence recognition model of the mathematical knowledge in math documents. If each sentence in math document is successfully recognized and expressed with this parallel phrase model and object-oriented model, eLearning system can answer various mathematical questions, e.g., "What is the definition of a connected graph?", "Show me an example of this theorem.", "Is it true that this assumption implies this formula?" and "Translate this proof into Japanese.".

Actually, we have just digitalized a book " Discreet Mathematics[5]" and extracted phrases and objects. We need extend the phrase dictionary and improve the knowledge extraction algorithms in the future.

.

References

[1] Hideto Ikeda: "Theoretical Foundation of International Common Language", Report of ICL study group" (in Japanese). Nobember, 2011.

[2] Hideto Ikeda, Nestor, SC and M. Ridwan. "Procedural Knowledge Management System", NLPKE-2011.

[3] Hideto Ikeda, Nguyen Thanh Hung and Yu Xinting: Interlingual Math Contents for eLearning.",

[4] M. Suzuki "Infty Project" http://www.inftyproject.org/en/software.html.

[5] Richard JphnsonBaugh: "Discrete Mathematics, Seventh Edition". Pearson Education Inc. 200

[6] Chomsky, Noam 1957. Syntactic Structures. The Hague/Paris: Mouton.

[7] Tesnière, Lucien 1959. Éleménts de syntaxe structurale. Paris: Klincksieck.

[8] Barkley Aligner. 2009. *A word alignment software package for machine translation*. http:// code. google. com/p / berkeleyaligner/

[9] Brundage, J. 2001. *Machine Translation – Evolution not Revolution*. Proc. MT Summit VIII Santiago

[10] Chiang D. 2007. *Hierarchical Phrase-Based Translation.* Computational Linguistics, Volume 33, Number 2, Association for Computational Linguistics.

[11]Frewley, William. 1988. *New Forms of Specialized Dictionaries. International Journal of Lexicography*. 1-3:189–213.

[12]Fontenelle, T. 1997. *Turning a Bilingual Dictionary into a Lexical-Semantic Database*. Tübingen: Niemeyer.

[13] Nagao, M. 1984. *A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, in Artificial and Human Intelligence*, A. Elithorn and R. Banerji (eds.) North- Holland, pp. 173-180.

[14] Nagao, M. 2003. *The Association for Computational Linguistics - 2003 ACL Lifetime Achievement Award*. Association for Computational Linguistics. http://www. aclweb.org/ index.php? option=com_content& task= view& id=36&Itemid=30. Retrieved 2010-03-10.

[15] NATools. 2010. *Workbench for parallel corpora processing*. http://linguateca.di.uminho.pt/natools/

[16] NiCT(National Institute of Information and Communications Technology). 2010. *Nict-EDR.*

[17] SNLPG (The Stanford Natural Language Processing Group) 2003. *Stanford Parser: A statistical Parser.* http://nlp.stanford.edu:8080/parser/index.jsp

[18] Wikipedia. 2010. *Machine translation*. http://en. wikipedia.org/wiki/Machine_translation