International Journal of Innovative Computing, Information and Control
Volume **14**, Number **1**, February **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 83–98

AUTOMATIC UNDERSTANDING AND FORMALIZATION OF NATURAL LANGUAGE GEOMETRY PROBLEMS USING SYNTAX-SEMANTICS MODELS

Wenbin Gan and Xinguo Yu

National Engineering Research Center for E-Learning
Central China Normal University
No. 152, Luoyu Road, Wuhan 430079, P. R. China
wenbingan@mails.ccnu.edu.cn

Received June 2017; revised October 2017

Abstract. *Automatic understanding of natural language problems is a long-standing challenge research problem in automatic solving. This paper models the understanding of geometry questions as a problem of relation extraction, instead of as the problem of semantic understanding of natural language; further it discovers that the entities and the geometric attribute pattern of elements can play an important role in relation extraction. Based on these ideas this paper proposes a syntax-semantics (S^2) model approach to understand geometry problem, targeting to produce a group of relations to represent the given geometry problem. The formalized geometric relations can then be transformed into the target system-native representations for manipulation to obtain geometric solutions. Experiments conducted on the test problem dataset show that 91.5% of questions can be correctly understood and solved, and the F_1 score in formalizing these problems is substantially high (0.990). The comparisons also demonstrate that the proposed method can achieve good performance against the state-of-the-art method. Integrating the automatic understanding method with different geometry systems will greatly enhance the efficiency and intelligence in automatic solving.*

Keywords: Understanding geometry problems, Formalized geometric propositions, Relation extraction, Syntax-semantics model, Automatic solution

1. Introduction.

Understanding problems described in natural language is a critical and challenging step of many automatic solvers [2, 3, 4, 12, 18, 19]. Developing automatic solvers of geometry problems in basic education has been a hot research problem due to the fact that it is a core technology in building intelligent educational systems that can provide step-by-step proofs for tutoring learners [24]. The research on the problem of understanding geometry problems in the basic education achieved good progress, but it is still an open research problem.

Currently, many geometry problem solving systems have been built to conduct automatic reasoning to get geometric proofs [4, 13, 14, 15, 17, 18, 25]. Ideally, these systems should understand the information presented in the natural language geometry problems themselves to extract the geometric relations among the elements and perform automatic solving by using some intelligent strategies [2, 3, 4, 10, 11]. However, due to the fact that the natural language description of a problem can be stated in various ways by different users, most geometry solving systems, to the best of our knowledge, perform the geometric reasoning and automatic solving based on the hypothesis that the natural language geometry problems are accurately understood and leave the problem understanding task to users. Specifically, these systems have a control of problem setups over the user-input through the predesigned input mechanism [13, 14, 17]. In other words, they cannot directly understand the geometry problems in a natural language environment. Some geometry solving systems incorporating algorithms for understanding geometry problems in natural language adopt the approach of semantic analysis [8, 9]. However, the semantic expressions of the same geometric relation have a slew of variants. Hence, the approach of deriving relations through semantic understanding requires a large number of models, even if all cases have their corresponding models.

This paper proposes a new approach to formalize and understand geometry problems to overcome the difficulty of understanding geometry problems through semantics analysis. These geometry problems are mainly plane geometry problems at secondary school level with no geometric quantities. As it is difficult for the existing solvers to deduce algebraic expressions involving geometric quantities because of the combinatorial explosion of search space [18, 20]. Informally, the proposed approach is to extract the geometric entities and geometric relations of these entities and constitute the geometric propositions, which is based on the idea that the elementary geometry deals with the relations of geometric entities. For achieving this aim, a set of syntax-semantics models are built to extract the geometric relations from the problem texts. Moreover, linguistic analysis in NLP area (e.g., syntax analysis) is used for parsing the geometry problem text to get the geometric entities including geometry elements and geometry relation words. A model matching algorithm is proposed to match the proper syntax-semantics model to extract geometric relations from each problem sentence using the geometric entities as indicators. This procedure is fully automated, a school-level geometry problem in Chinese language is understood and transformed into formalized geometric propositions in the form of first-order predicate logic. These formalized propositions can be further written in the target system-native representation for direct manipulation and used for various tasks. Experiments conducted on a geometry problem dataset show the effectiveness of the proposed approach.

The proposed approach has multiple merits. Firstly, the problem understanding is converted into relation extraction. Informally, it models the automatic understanding as a problem of the finite pattern recognition that overcomes the innumerous varieties of semantic meanings of natural language. Secondly, relation extraction is converted into syntax-semantics model matching, which is a more easily executive procedure in pattern matching.

This paper is organized as follows. Related work is presented in Section 2. The problem definition of geometry problem understanding is described in Section 3. Section 4 presents the technical details of the proposed approach. Analysis and illustration of the experimental results are presented in Section 5 and finally the paper is concluded in Section 6.

2. **Related Work.**

Many geometry systems with deductive mechanism were built with focus on geometry theorem proving and dynamic diagram construction, such as Geometry Expert [13], GeoProof [14], GEOTHER [15], Cinderella [16] and GeoGebra [17]. In these systems, geometry problems and conjectures are input either following construction in the point-and-click manner, or as formalized geometric statements unfamiliar to typical users. None of the systems involve machines to understand and formalize the natural language [10, 11] which is a common form of problem description in geometry domain.

To solve this problem, Liu et al. used geometric ontology and relation pattern base to transform the restricted problem texts into command sequences [7]. They discovered the geometric elements and their relations in the problem text and represented each relation as an n-triple. These n-triples are matched with the predefined relation patterns to obtain the command sequences. The geometric relations are extracted from the whole problem text rather than separate sentences, thus making the transformation inefficient and error prone.

Some researchers conducted automatic problem understanding to ground geometry problem texts into underlying relation sets by the aid of specific knowledge base. Wong et al. used a cognitive knowledge base named InfoMap to extract the exact category of a geometry problem and the problem-concept set using template matching mechanism and formalized a problem as a concept-attribute content tree containing hierarchical n-odes of problem category, problem concepts and linguistic knowledge [2]. The knowledge framework is predefined and the problems solvable are certain types of geometric shape problems concerning perimeter and area of elementary shapes. Thereby it is difficult to formalize the problems containing complex geometric constructions and multiple categories. Mukherjee et al. used a knowledge base called GeometryNet [6] to interpret the geometric meaning of an input text to diagram descriptions [3, 5]. They decomposed the extracted entities into atomic entities by consulting the concepts in GeometryNet and used connecter to link the entities to form a parse graph. The graph is then translated into formal representation. The intermediate graph representation is more suitable for the aim of geometric construction but is difficult to use for other aims like problem solving.

Seo et al. mapped a multi-choice geometry problem into logical representation by interpreting both problem text and diagram. They formalized the problem sentences into hypergraph representations and used a discriminative model to measure the interpretation score of a relation between the concepts in hypergraph [4]. The combination with diagram interpretation remedies the incorrect understanding in the over-generated logical formulas from problem text. By following the idea of mapping and grounding the natural language descriptions to specific forms, some researchers adopted the transformation ideology [26, 27] to formalize the geometric statements in natural language. Chen proposed a framework to automatically transform the geometric statements to propositions using the geometry description language (GDL) [1]. Based on the syntax of GDL, concept matching and transforming rules are presented to transform a geometric statement into equivalent statement in terms of basic concepts. However, the syntaxes of GDL and the rules used for formalizing statements are so complicated that the problems could only be transformed manually in the current stage, and the automated method of translating natural statements into GDL statements is not implemented at present.

The state-of-the-art method for automated understanding of geometry problem in natural language is sentence-template based method, which is used in some geometry systems that provide the natural language interactive interface. This method uses predesigned sentence templates to understand a problem sentence. The sentence templates consist of a series of variables and keywords in specific order and the matching process is sequentially executed by comparing the corresponding items in the sentence and each template. If a template matches with the sentence, the useful information in the sentence will be extracted. Following this method, [8, 9] designed various geometry sentence templates and used template matching to extract the contained geometric relations from the problem text. However, the templates designed are in large numbers but are still incomplete to process the innumerous varieties of semantic meanings of natural language. The results they got have a high precision but a very low recall in relation extraction and the procedure in designing these templates is quite complicated and demanding.

Our work is related with the sentence-template based method in relation extraction but significantly different from the method in two important aspects. Firstly, instead of matching a sentence model using all the characters and syntax information in the sentence incorporating some relation-irrelevant information, we only use the directly related geometric entities to match the syntax-semantics model, thus making the relation extraction algorithm more flexible. Secondly, the sentence templates are incomplete for understanding the natural language geometry problems since the large volume of varieties of semantic meanings, comparatively, the number of geometric relations in elementary geometry is very limited. Hence the syntax-semantics models used in the proposed method are much complete in extracting the geometric relations and it also significantly decreases the number of required models.

3. **Problem Definition.**

This section formulizes the problem of understanding geometry problems. Given a school-level geometry problem in natural language, the goal of this study is to automatically understand and formalize the problem. This paper gives a descriptive definition of problem understanding for the geometry problems at secondary school level as below.

The problem of understanding geometry problems does not have a general and formal definition yet, though a lot of papers have addressed the automatic solving of such problems. This paper proposes a new approach for automatically understanding geometry problems. It targets to get a set of geometric relations.

Definition 3.1. (Equivalent representation): *A group of relations is called an equivalent representation of a given geometry problem if an algorithm can produce the solution of the given problem from this set of relations without revisiting the given problem.*

Definition 3.2. *Problem understanding is to produce a group of relations that is an equivalent representation of a given geometry problem.*

Under these two definitions, the objective of geometry problem understanding is to produce a set of relations to equivalently represent the geometry problem. Given the geometry problem text T, the understanding of textual information is to identify a set of geometry elements $E = \{E_1, E_2, \dots, E_n\}$ from T, and find the geometric relations $R = \{R_1, R_2, \dots, R_i\}$ among the set E.

Therefore, the understanding of geometry problem is converted into relation extraction from the problem text. How to extract the relations in the problems is critical to successfully implement the proposed approach. General natural language understanding targets to understand the semantic meaning of text. However, a geometry relation can have a slew of semantic expressions. A lot of models are needed if the semantic understanding approach is adopted. Hence, the approach of extracting relations through semantic understanding is not practical. This paper discovers that the relations can be extracted by using a pool of syntax-semantics (S^2) models, in which the syntax portions are the patterns of geometric types of elements and the semantic portions are keyword structures of geometric relation. The existing software can parse the problem text in natural language into phrases and label the part-of-speech of phrases with a high accuracy [23]. This approach is general to multiple nature languages. However, this paper presents the algorithm for understanding geometry problems in Chinese as example.

The extracted relations are represented as a set of atomic propositions using first-order predicates. The predicates here can be classified into three categories:

- Geometric element, such as *parallelogram (ABCD)*;
- Position relation, such as *midpoint (E, BC)*;
- Quantity relation, such as *eqAngle (ABC, DEF).*

By using the S^2 models, the geometry problems are understood and formalized into a set of atomic propositions. These formalized propositions can be further used for manipulation in various tasks.

4. **Our Approach.** This section is to present our approach to automatically extract a set of atomic propositions that can represent a given geometry problem. The framework of the proposed approach is shown in Figure 1. Here we first give an overview of the approach and then detail respective components.

Given a school-level plane geometry problem in natural language, our approach mainly uses three steps to output a set of relations of being equivalent to the given problem in finding solution.

- **Step 1 (Parsing and annotation)**: Uniform the problem text; parse the problem text into phrases and annotate each phrase with part-of-speech (POS) labels; perform sentence boundary detection to divide the text into sentences.
- **Step 2 (Entity identification)**: Extract geometric entities from each sentence, and then recognize the types of the extracted entities. All the results of extraction and recognition for a sentence form an annotation set ω;
- **Step 3 (Relation extraction)**: Use the syntax-semantics models to extract the relation from each sentence according to its annotation set ω.

To improve the accuracy of formalized results, preprocessing of the natural language statement and relation completion of the formalized geometric propositions are also conducted. Specific descriptions of the respective components are in the following subsections.

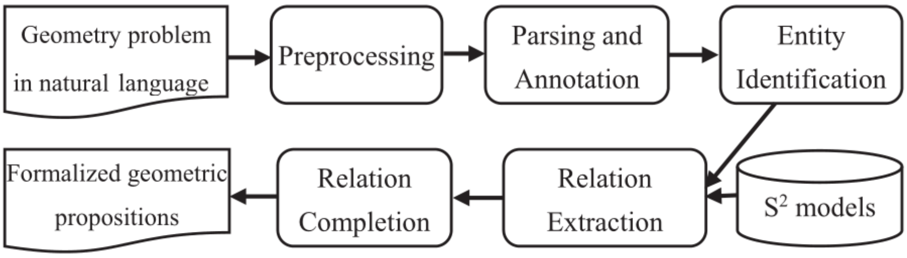
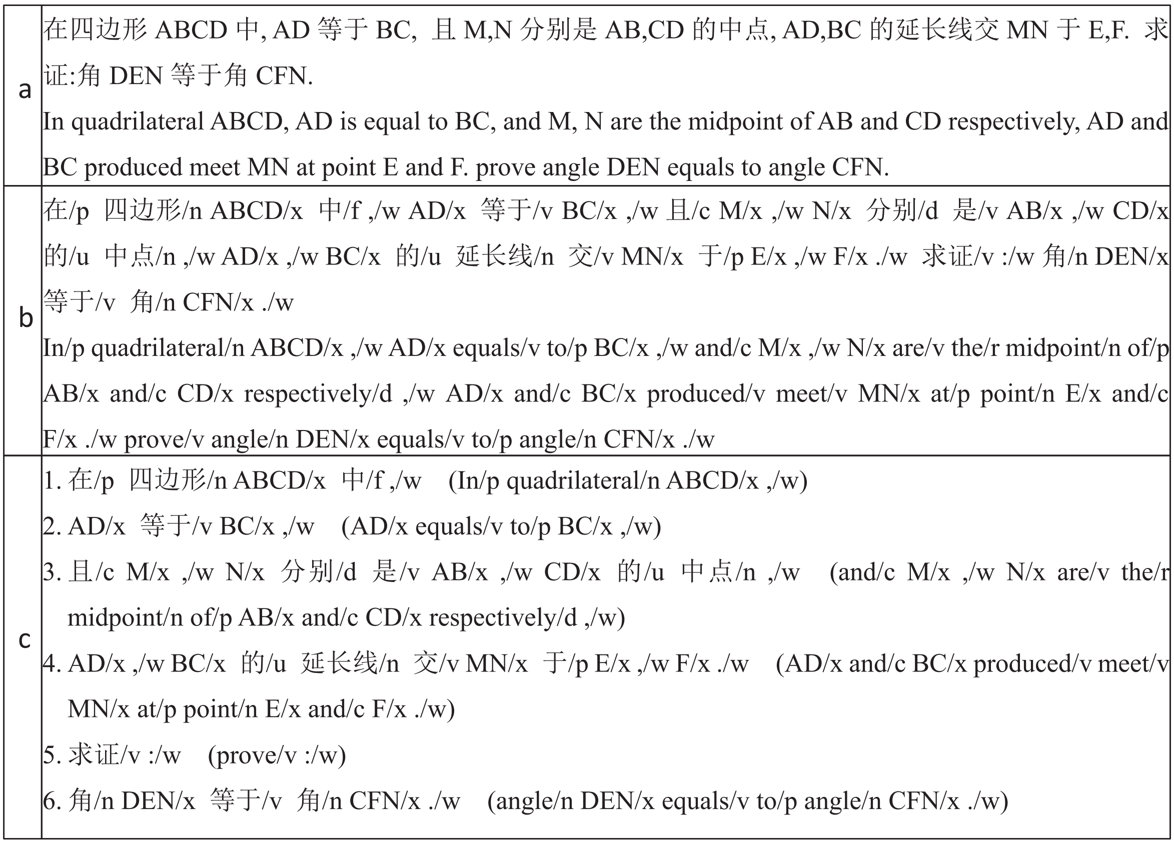


FIGURE 1. The framework of the proposed approach to formalize and understand a geometry problem

4.1. Parsing and annotation.

4.1.1. *Preprocessing.* The geometry statement in Chinese usually contains geometric symbols and format information. These particular symbols and information make it difficult for the following syntax analysis. Hence the input statements are normalized in three aspects. Firstly, the full-width letters and numbers are transformed into half-width ones, and the Chinese punctuation symbols into the corresponding symbols in English. Secondly, some particular geometric symbols are replaced by the corresponding Chinese description words. For example, "≅" is replaced by the word "*congruent*". Thirdly, the format information is removed, such as line feeds and multiple space. After preprocessing, the geometry statement contains several simple word types: Chinese characters, English letters, numbers and punctuation symbols.

4.1.2. *Parsing and annotation.* Chinese is generally written without word boundaries. To extract the entities from a geometry problem with several sentences, problem text must undergo word segmentation and POS tagging. We use an NLP tool named ICTCLAS [23], a perfect Chinese word segmentation system with the accuracy of 98.345%, to tackle this task. In order to improve the accuracy in parsing the sentences in geometry domain, a geometric dictionary is used as the user dictionary. An example problem is illustrated in Figure 2(a) and the result of segmentation and tagging is shown in Figure 2(b). A



a	在四边形 ABCD 中, AD 等于 BC, 且 M,N 分别是 AB,CD 的中点, AD,BC 的延长线交 MN 于 E,F. 求证:角 DEN 等于角 CFN. In quadrilateral ABCD, AD is equal to BC, and M, N are the midpoint of AB and CD respectively, AD and BC produced meet MN at point E and F. prove angle DEN equals to angle CFN.
b	在/p 四边形/n ABCD/x 中/f ,/w AD/x 等于/v BC/x ,/w 且/c M/x ,/w N/x 分别/d 是/v AB/x ,/w CD/x 的/u 中点/n ,/w AD/x ,/w BC/x 的/u 延长线/n 交/v MN/x 于/p E/x ,/w F/x ./w 求证/v :/w 角/n DEN/x 等于/v 角/n CFN/x ./w In/p quadrilateral/n ABCD/x ,/w AD/x equals/v to/p BC/x ,/w and/c M/x ,/w N/x are/v the/r midpoint/n of/p AB/x and/c CD/x respectively/d ,/w AD/x and/c BC/x produced/v meet/v MN/x at/p point/n E/x and/c F/x ./w prove/v angle/n DEN/x equals/v to/p angle/n CFN/x ./w
c	1. 在/p 四边形/n ABCD/x 中/f ,/w (In/p quadrilateral/n ABCD/x ,/w) 2. AD/x 等于/v BC/x ,/w (AD/x equals/v to/p BC/x ,/w) 3. 且/c M/x ,/w N/x 分别/d 是/v AB/x ,/w CD/x 的/u 中点/n ,/w (and/c M/x ,/w N/x are/v the/r midpoint/n of/p AB/x and/c CD/x respectively/d ,/w) 4. AD/x ,/w BC/x 的/u 延长线/n 交/v MN/x 于/p E/x ,/w F/x ./w (AD/x and/c BC/x produced/v meet/v MN/x at/p point/n E/x and/c F/x ./w) 5. 求证/v :/w (prove/v :/w) 6. 角/n DEN/x 等于/v 角/n CFN/x ./w (angle/n DEN/x equals/v to/p angle/n CFN/x ./w)

FIGURE 2. An example geometry problem and the result of parsing and annotation. (a) An example geometry problem; (b) the result of word segmentation and POS tagging; (c) the result of sentence boundary detection, each line is a separate sentence.

geometric relation is usually explicitly contained in one sentence; hence the geometry problem should be divided into several simple sentences. Here a rule-based sentence boundary detection method is adopted to judge whether a test line should be broken into two. A group of words followed by a comma (,), a semicolon (;), a period (.) and a colon (:) are labeled as a sentence. Several cases arise when connectors like "," appear in a geometry statement:

1) If a "," separates two geometry elements (tagged with "/x") of the sample type, the division should not take place. For example, in the sentence "*M, N are the midpoint of AB, CD*", the comma between "M" and "N" (or "AB" and "CD") separates two geometry elements of the type "point" (or "line segment"), and the sentence is not broken into two.
2) If a "," separates two different types of geometry elements, the sentence should be divided into two short ones. For example, "*line EC intersects DA at F, AE is equal to AF*". Here, the comma lies between two different types of elements ("F" is the type of "point" while "AE" is the type of "line segment"), so the division should take place.

4.2. **Entity identification.** After conducting sentence boundary detection, a tagged problem is divided into separate sentences which contain the geometric relations to be extracted. The division result of the problem in Figure 2(a) is illustrated in Figure 2(c). Then two categories of geometric entities constituting the geometric relations are extracted in each sentence. The first one is geometry element and the other is the geometry relation indicator.

Definition 4.1. (Geometric element representation): *A geometric element representation is a duple $e = (w, t)$ in which w is a phrase, and t is the geometry type of w. The types of geometric elements include points, lines, triangles, angles and some special geometric shapes.*

This paper has identified 48 kinds of geometric relations and each relation is named after a relation word. Examples of relation words are "Parallel" and "MidPoint". Each of these geometric relations has a collection of variant expressions in problem text. A table is created to include all these relation words and their expression variant representations.

Definition 4.2. (Geometric relation representation): *A geometric relation representation is a duple $J = (v, o)$ in which o is a representative relation word and v is the variant list of o.*

We discover that the POS tags are helpful in extracting the geometric entities. As shown in Figure 2(b), all the geometric elements are tagged with "/x", and most of the words with tag "/v" or "/n" are the relation words. For each sentence, we extract these entities and get a list of geometric elements and relation words. Some words tagged with "/v" or "/n" but not belonging to the relation words, such as "extension line", "draw", "suppose", "is" and "prove", are labeled as stop words and removed from the list of relation words. Following this method, two lists E and J of geometric entities are extracted. The entities identified in each sentence of Figure 2(c) are shown in Figure 3. It is worth noting that the type of a geometric element is assigned using domain knowledge based on the number of capital letters. This works because if a specialized entity were to use in a problem, the type should be explicitly mentioned. For example, in the sentence "AB is equal to CD", it is easy to infer that AB and CD are both line segments. However, if ABC is used in a sentence, it has to be mentioned whether it is an angle or a triangle.

Sentence NO.	Geometric element	Geometric relation word
1	(ABCD, quadrilateral)	(四边形, quadrilateral)
2	(AD, line), (BC, line)	(等于, eqDistance)
3	(M, point), (N, point), (AB, line), (CD, line)	(中点, midpoint)
4	(AD, line), (BC, line), (MN, line), (E, point), (F, point)	(交, intersect)
6	(DEN, angle), (CFN, angle)	(等于, eqAngle)

FIGURE 3. The entities extracted in each sentence of Figure 2(c). Note that the 5th sentence contains stop words and is removed.

4.3. **Relation extraction.** Relation extraction is a key step for transforming the natural language sentences of a geometry problem to formalized geometric propositions by using the predefined syntax-semantics (S^2) models.

Definition 4.3. (S^2 model): *An S^2 model for plane geometry problems is defined as a triple $M = (J, E, F)$, where J represents the geometric relation representation, $E = \{e_1, e_2, e_3\}$ is the set of the involved elements, and F is the atomic proposition in first order predicate logic (FOL). Let $\Pi = \{M_i = (J_i, E_i, F_i) | i = 1, 2, \ldots, n\}$ denote all the prepared S^2 models. It is also called as a pool of S^2 models of plane geometry.*

The syntax portion of an S^2 model is the change pattern of types of geometric elements and the semantic portion is keywords of geometric relation. The types of geometric elements include points, lines, triangles, angles and some special geometric shapes. The keywords of semantic portion include the 48 kinds of geometric relation words and their variants. A total of 48 kinds of geometric relations are identified. Each S^2 model contains one geometric relation so that there are 48 S^2 models.

The 48 kinds of geometric relations in plane geometry can be divided into three types, namely, unary, binary and ternary relation, as shown in Table 1. Each such relation corresponds to an atomic proposition so that there are 48 atomic propositions.

Table 1. Explanation of three types of geometry relations

Type	Feature	Meaning	FOL	#
Unary	(ABC, equilateral_triangle)	ABC is a equilateral triangle	eqTriangle (ABC)	17
Binary	(AB, CD, line, line)	line AB is parallel to CD	parallel (AB, CD)	22
Ternary	(AB, CD, E, line, line, point)	line AB intersects CD at point E	intersect (E, AB, CD)	9

The pool of S^2 models is used to extract the atomic propositions, as described in Algorithm 1.

Algorithm 1: Extraction of geometry relations using S^2 models

```
Input: a set of simple sentences of a plane geometry problem $T$, each sentence $S$ is
       annotated with its geometry element representation $E'$ and geometry relation
       representation $J'$.
Output: the contained atomic propositions in each sentence, denoted as $R$.
Load S$^2$ models $\Pi = \{M_i = (J_i, E_i, F_i) | i = 1, 2...n\}$;
Initialize $R$ as empty;
while TRUE do
    Pick a simple sentence from the sentence set;
    for i from 1 to n do
        if match $J_i$ of $M_i$ with $J'$ is FALSE then
            Continue;
        end
        if match the number and types of $E_i$ with $E'$ is FALSE then
            Continue;
        end
        Put the instantiated $F_i$ of $M_i$ into $R$;
    end
    if all sentences are processed then
        break While loop;
    end
end
```

As shown in Algorithm 1, each sentence of a geometry problem is annotated with its geometry entities including the geometry element representation and geometry relation representation using the method described in Sections 4.1 and 4.2. The geometric relations are extracted sentence by sentence. The entities in a sentence are used as indicators for matching the proper S^2 models. The geometry relation representation is first used to match the geometric relation word in an S^2 model. If it is matched then the number and types of geometric elements in the geometry element representation are tested with the element part in the S^2 model. Only when all these conditions are matched, an S^2 model will be activated for a sentence. The geometric elements in the sentence will be

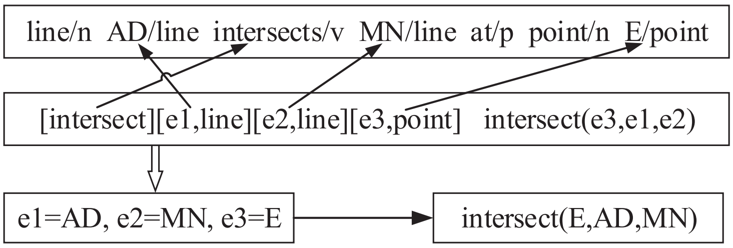

line/n AD/line intersects/v MN/line at/p point/n E/point
[intersect][e1,line][e2,line][e3,point] intersect(e3,e1,e2)
e1=AD, e2=MN, e3=E
intersect(E,AD,MN)

FIGURE 4. Atomic geometric relation extraction using S^2 model for a geometry problem sentence

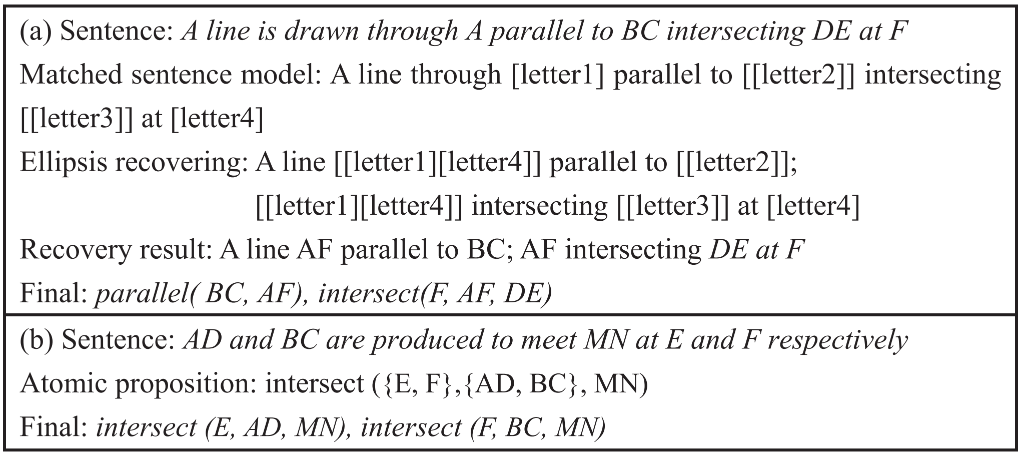
extracted. And an atomic proposition will be generated by applying the predicate in the model to a sequence of matched geometric elements. An example of relation extraction using Algorithm 1 is shown in Figure 4. Repeating this process for each sentence of a geometry problem, eventually a set of atomic propositions will be obtained. These atomic propositions are equivalent to the given geometry problem in the sense of finding solutions.

It is worth noting that a given Chinese keyword may have two different predicates. For example, the word perpendicular matches with the predicate "perpendicular" and "foot" in the sentence "*AB is perpendicular to CD*" and "*AB is perpendicular to CD at point F*" respectively. In such cases, the number of geometric elements and their types are used to differentiate the appropriate models.

Relation completion. So far, we have explained how to automatically formalize the question sentences into propositions. This is effective when a sentence only contains one explicit geometric relation, but geometry problems usually include implicit concepts and may have more than one geometric relations in each sentence. In addition, ellipses and coordinate structures are frequently used in stating a geometry problem [4]. Ellipses usually happen when two or more geometric relation words exist in one sentence, while coordinate structures always occur when several adjacent geometric elements are separated by "," or "and". In "*A line is drawn through A parallel to BC intersecting DE at F*", the entity mention of the line ("*AF*") paralleling to *BC* is omitted. Also, consider the following sentence "*AD and BC are produced to meet MN at E and F respectively*". Here, "*AD, BC*" and "*E, F*" are coordinate structures.

It is difficult to directly use the S^2 model matching algorithm (Algorithm 1) to get the right geometric relations because of the ellipses of geometric entities and the over-numbered geometric elements in coordinate structures. For the case of ellipses, entities are recovered by using sentence template method before the relation extraction. Figure 5(a) shows how a sentence model is matched to recover the entity mention "AF" in the example sentence used in the previous paragraph. Note that this sentence contains two geometric relation and is broken into two simple ones. These two simple sentences are processed by Algorithm 1 to get the final formalized propositions. For the case of coordinate structures, the geometric elements which are coordinate are written in a pair of brackets ("{}") in the formalized proposition and then are assigned to separated propositions after the relation extraction. Figure 5(b) shows the processing of an example containing coordinate structures.

After relation extraction and relation completion, the problem in Figure 2(a) is formalized to a set of atomic geometric propositions shown in Section 5.3.1.



(a) Sentence: *A line is drawn through A parallel to BC intersecting DE at F*

Matched sentence model: A line through [letter1] parallel to [[letter2]] intersecting [[letter3]] at [letter4]

Ellipsis recovering: A line [[letter1][letter4]] parallel to [[letter2]];
[[letter1][letter4]] intersecting [[letter3]] at [letter4]

Recovery result: A line AF parallel to BC; AF intersecting *DE at F*

Final: *parallel(BC, AF), intersect(F, AF, DE)*

(b) Sentence: *AD and BC are produced to meet MN at E and F respectively*

Atomic proposition: intersect ({E, F},{AD, BC}, MN)

Final: *intersect (E, AD, MN), intersect (F, BC, MN)*

Figure 5. The relation completion of geometry sentences incorporating two cases. (a) The template matching method of ellipsis recovering; (b) the processing of formalized proposition with coordinate structures.

5. Experimental Evaluation.

5.1. Dataset.

In order to evaluate the effectiveness of the proposed method, a dataset of 162 plane geometry proof problems was built. The problems are collected from the PEP edition of Chinese mathematics textbooks for junior high school students of grades 8 and 9 and a professional book [22] in plane geometry and some problems in [21]. In addition, a portion of the publicly available plane geometry problems for senior high schools are also used. We collect mainly those problems that do not mix algebraic expressions or computations in the problem stems. The texts of these problems are understood by people and manually input to the geometry theorem proving system Java Geometry Expert (JGEX) [13] by interactively drawing dynamic diagrams. Eventually 130 geometry problems can be solved by using the provided proving methods, and these problems are used to form the test dataset. For each problem we manually prepare a set of atomic propositions as its groundtruth. Table 2 gives the statistics of the problems and the groundtruth of the dataset.

Table 2. Statistics on the problems and the groundtruth of test dataset

#	Statistics on problems			Statistics on groundtruth			
	Questions	Sentences	Words	UR	BR	TR	Total propositions
Total	130	685	3862	202	392	191	785
Average	1	5	30	2	3	2	6

Note: UR =: Unary relation, BR =: Binary relation, TR =: Ternary relation.

5.2. Experimental setup.

5.2.1. *Syntax-semantics model building.* After analyzing all the geometric concepts and geometric relations appearing in the PEP edition of Chinese mathematics textbooks for junior high school students of grades 7, 8 and 9, a total of 48 syntax-semantics models are built and stored in a relation model database.

5.2.2. *Baseline method.* The method presented in [9] is selected as a baseline method for comparing with the proposed method because it is the state-of-the-art method that addresses the understanding of plane geometry problems. The algorithm in [9] uses sentence template matching (STM) method to convert the problem in Chinese into the problem in restricted language. STM method understands the geometry problems by using the predefined sentence templates. If a sentence template is matched with a sentence in a problem, the relations contained in the sentence will be output. In [9], they prepared a total of 196 sentence templates.

5.2.3. *Evaluation.* Two aspects are evaluated in the experiment. Firstly, the performance of solving geometry problems. A geometry problem understanding and transformation system is developed to understand the problems. The result of problem understanding is input to the geometry solving system for getting solution. The number of solved problems can reflect the capability of the proposed method in problem understanding. Secondly, the performance of relation extraction is evaluated between the proposed method and the baseline method. Precision, recall and F$_1$-measure are used to reflect the performance. Assuming that a method extracts m of n geometric relations in a test set and k of m are correct. (1) $P\,(precision) = k/m$, (2) $R\,(recall) = k/n$, (3) $F_1 = 2k/(m+n)$.

5.3. **Experimental results.**

5.3.1. *Solving geometry problems.* A geometry problem understanding and transformation system (GPUTS) is developed and integrated with JGEX to unite the two systems for dynamic diagram drawing and problem solving. Figure 6 (left) shows the interface of GPUTS. Each test problem is input to the system and the geometric relations in the problem text are extracted and formalized to atomic propositions in the predefined format. The middle of the interface shows the formalized propositions and the matched models are also listed. These propositions are then transformed into the native language of JGEX. Then these transformed propositions are imported into JGEX and a dynamic diagram can be automatically generated. Moreover, specific solving method, for example, the deductive database method and Wu's method, can be selected to generate the geometric proofs.

An application to the example problem in Figure 2 (in Section 4) is also shown in Figure 6. After problem understanding and formalization, the problem is transformed into a geometry proposition shown as follows.

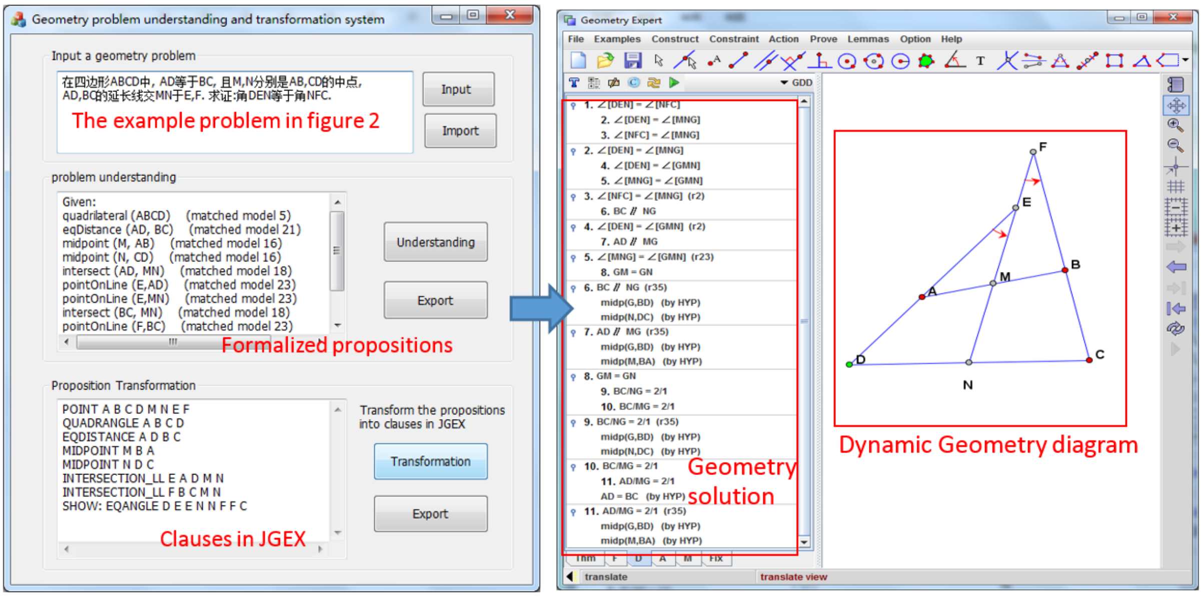
Given: quadrilateral (ABCD), eqDistance (AD, BC), midpoint (M, AB), midpoint (N, CD), intersect (E, AD, MN), intersect (F, BC, MN).

Prove: eqAngle (DEN, CFN).

The proofs of solving the example problem using deductive database method are shown in Figure 6 (right).

Here we choose JGEX as an example of geometry solving system for the following two reasons. Firstly, it is a famous geometry system widely used for dynamic diagram drawing, visually dynamic presentation of proofs and automated geometry theorem proving and discovering and it is also free to use[1]. Secondly, the formalized atomic propositions are easy to be transformed into the clauses in JGEX. According to the rules in Table 3, geometric predicates are mapped to the corresponding concepts in JGEX and geometric elements only change in form but not in content. This translation is easy to be achieved because these propositions reserve the original concepts in the problem statement and do not need any semantic translation of the involved concepts.

[1] The software of JGEX can be downloaded from here: http://www.cs.wichita.edu/ye/.



在四边形ABCD中，AD等于BC, 且M,N分别是AB,CD的中点,
AD,BC的延长线交MN于E,F. 求证:角DEN等于角NFC.

Given:
quadrilateral (ABCD) (matched model 5)
eqDistance (AD, BC) (matched model 21)
midpoint (M, AB) (matched model 16)
midpoint (N, CD) (matched model 16)
intersect (AD, MN) (matched model 18)
pointOnLine (E,AD) (matched model 23)
pointOnLine (E,MN) (matched model 23)
intersect (BC, MN) (matched model 18)
pointOnLine (F,BC) (matched model 23)

```
POINT A B C D M N E F
QUADRANGLE A B C D
EQDISTANCE A D B C
MIDPOINT M B A
MIDPOINT N D C
INTERSECTION_LL E A D M N
INTERSECTION_LL F B C M N
SHOW: EQANGLE D E E N N F F C
```

1. ∠[DEN] = ∠[NFC]
    2. ∠[DEN] = ∠[MNG]
    3. ∠[NFC] = ∠[MNG]
2. ∠[DEN] = ∠[MNG]
    4. ∠[DEN] = ∠[GMN]
    5. ∠[MNG] = ∠[GMN]
3. ∠[NFC] = ∠[MNG] (r2)
    6. BC // NG
4. ∠[DEN] = ∠[GMN] (r2)
    7. AD // MG
5. ∠[MNG] = ∠[GMN] (r23)
    8. GM = GN
6. BC // NG (r35)
    midp(G,BD) (by HYP)
    midp(N,DC) (by HYP)
7. AD // MG (r35)
    midp(G,BD) (by HYP)
    midp(M,BA) (by HYP)
8. GM = GN
    9. BC/NG = 2/1
    10. BC/MG = 2/1
9. BC/NG = 2/1 (r35)
    midp(G,BD) (by HYP)
    midp(N,DC) (by HYP)
10. BC/MG = 2/1
    11. AD/MG = 2/1
    AD = BC (by HYP)
11. AD/MG = 2/1 (r35)
    midp(G,BD) (by HYP)
    midp(M,BA) (by HYP)

FIGURE 6. (left) The interface of GPUTS; (right) dynamic diagram construction and problem solving in JGEX

TABLE 3. Mapping the formalized propositions into clauses in JGEX

Formalized proposition	Corresponding clause in JGEX
quadrilateral(ABCD)	QUADRANGLE A B C D
eqDistance(AD, BC)	EQDISTANCE A D B C
midpoint(M, AB)	MIDPOINT M A B
intersect(E, AD, MN)	INTERSECTION_LL E A D M N
eqAngle(DEN, CFN)	EQANGLE D E N C F N
parallel(AB, CD)	PARALLEL A B C D
foot(E, AB, CD)	FOOT E A B C D
bisect(AG, BAC)	ANGLE_BISECTOR G B A C
......	......

We evaluate the effectiveness of the proposed method using the number of solved geometry problems in test dataset. This is reasonable because if a geometry problem is not correctly understood, the problem may be not solved to a large extent. Table 4 shows the results of the problem solving experiment. 91.5% of test problems can be correctly solved using the proposed method, and 11 problems cannot be solved because of defective configurations caused by the faulted and omissive formalization. Error analysis of these unsolved problems will be discussed in Section 5.3.3. As a comparison, the baseline method using sentence template matching solves 73.1% of test problems. These unsolved problems are mainly because of omissive formalization. This result shows that the proposed method has better capability in problem understanding than the state-of-the-art method and the automatic understanding can enhance the effectiveness and intelligence of existing geometry solvers.

5.3.2. *Performance of relation extraction.* We further evaluate the performance of relation extraction to better understand why the proposed method performs better in problems understanding. Table 5 details the comparison on performances of extracting three kinds of geometric relations between the proposed and the baseline methods. The proposed

Table 4. The comparison of numbers of test problems solved and unsolved using two methods

	Proposed method		Baseline method	
	Solved	Unsolved	Solved	Unsolved
Number of questions	**119**	11	95	35
Percentage	**91.5%**	8.5%	73.1%	26.9%

Table 5. Comparison on performances of extracting three kinds of geometric relations between the proposed and the baseline algorithms

	Proposed method			Proposed method w/o RC			Baseline method		
	P	R	F1	P	R	F1	P	R	F1
Unary relation	**1.0**	0.990	**0.995**	1.0	0.990	0.995	1.0	0.772	0.871
Binary relation	0.997	**0.995**	**0.996**	0.917	0.898	0.907	0.978	0.673	0.797
Ternary relation	0.979	0.963	**0.971**	0.736	0.702	0.719	0.960	0.634	0.764
Total	0.994	0.986	**0.990**	0.896	0.874	0.885	0.98	0.689	0.809
Model number			48					196	

method attains substantially high F_1 score (0.990) in extracting relations from the texts of geometry problems by using the 48 S^2 models. Unary relation attains the best precision (1.0) and binary relation has the best recall (0.995), while ternary relation performs less perfect than the other two but is still quite good.

Ablation test is also conducted without using the relation completion (w/o RC) to understand its effectiveness. As shown in Table 5, the full method outperforms the ablation method both for binary and ternary relation extraction while having no improvement in unary relation extraction. It can be also noticed that the RC component is more effective to the ternary relation than to the binary relation extraction. It improves performance by 25.2% in F1 score for ternary relation and 8.9% for binary relation. This is consistent with the trait of test questions as there are more ellipses, coordinate structures belonging to ternary geometric relation than binary relation in problem text. This result verifies that the relation completion procedure is effective for the extraction of geometric relations contained in the problem text and it benefits the performance of problem understanding.

By contrast, the proposed method outperforms the baseline method in extracting relations. The baseline method also achieves quite high precisions in extracting the three kinds of relations but at the cost of lower recalls. This is reasonable at two aspects in terms of method. Firstly, the sentence templates in the baseline method consist of a series of variables and keywords in specific order and the matching process is sequentially executed by comparing the corresponding items in the sentence and the template. While the syntax-semantics models used in the proposed method are centered on the geometric relations themselves. And it improves the matching strategy by only using the useful information that forms a geometric relation like the relation word itself and the number and types of geometric elements to match the relation models and takes no regard for the various statement forms of natural language. For example, "*AB intersects CD at the point E*" and "*E is the intersection of AB and CD*". To extract the relation in these two sentences, more sentence templates are needed while no more S^2 models need to be added. The proposed method extracts the geometric entities and uses them to match the syntax-semantics model without considering the specific orders of involved elements. Secondly, sentence templates are incomplete for all the geometry problems (although a total of 196 sentence templates are used) while the syntax-semantics models are relatively complete for the elementary geometry problems. Since there are innumerous varieties of semantic meanings of natural language, a single template should be added to match a kind of statement form. Comparatively, the number of geometric relations in elementary geometry is very limited (in our experiment a total of 48 common geometric relations are found). Each relation is incorporated in one syntax-semantics model. The results in Table 5 also verify that the proposed method obtains better performance and can also significantly decrease the number of required models.

5.3.3. *Error analysis.* In order to understand the formalization errors made by the proposed method, 11 unsolved geometry problems are analyzed. The formalization errors can be divided into two categories: faulted formalization and omissive formalization. Table 6 details the error distribution in extracting three kinds of relations. Roughly 64% of omissive formalization errors and 80% of faulted formalization errors are made during the formalization of geometry sentences containing ternary relations. Part of the examples of these two kinds of errors and the remedial measures adopted are listed as below.

(1) Faulted formalization error. In the sentence "*draw a line BM perpendicular to EF through point B*", the formalization result is "*foot(EF, BM, B)*". In fact, the foot point is M rather than B although point B is explicitly stated. To solve this issue, corresponding sentence models are used to normalize the sentences to help extract the correct geometric elements before the relation extraction.

(2) Omissive formalization error. Considering the sentence "*in the triangle ABC, H is the orthocenter*". After sentence boundary detection, the sentence is divided into two short ones. In such a case, the system does not know whose orthocenter the point H is and this may cause omissive formalization. Also consider another example "*Two altitudes AD and BE intersect at point H*". There are three geometric relations in this sentence: two altitudes AD and BE and their intersection H. After relation extraction, the intersection relation *intersect (H, AD, BE)* may be output, but the altitude relations may be omitted. This is due to the reason of not knowing whom the altitudes belong to. However, based on the statement paradigms of geometry problem, a geometric entity is usually stated first and then the parts of it are introduced subsequently. Therefore, omissive formalization errors can be revised by backtracking to the previous sentence to extract the appropriate geometric entities and adding the omitted geometric relations to the formalized propositions.

Table 6. The distribution of two kinds of errors in transforming three relations

	Faulted formalization	Omissive formalization
Unary relation	0	2
Binary relation	1	2
Ternary relation	4	7

6. **Discussions and Conclusions.** This paper has presented a new approach to automatically understand and formalize geometry problems in natural language. The contributions of our work are mainly embodied in three aspects. Firstly, it proposes a new approach of understanding geometry problems, which is to extract a set of geometric relations to equivalently represent the given problem in terms of finding solution. This approach models the problem understanding as a problem of the finite pattern recognition that overcomes the innumerous varieties of semantic meanings of natural language. Secondly, it proposes a syntax-semantics (S^2) model method to extract the relations. This

paper discovers and verifies that the geometric relations can be extracted by using a pool of S^2 models, in which the syntax portions are the patterns of geometric types of elements and the semantic portions are keywords structures. Thirdly, a relation extraction algorithm is proposed and a geometry problem understanding and transformation system integrating with JGEX is developed. Natural language geometry problems can be input to the system and transformed into the formalized propositions. As an intermediate representation, these propositions can be then transformed into the target system-native representations for manipulation in various tasks. Experiment conducted on the test problem dataset shows that 91.5% of problems can be correctly understood and solved, and the F$_1$ score in formalizing these problems is substantially high (0.990). The comparison with state-of-the-art method also verifies the effectiveness of the proposed method. Employing the proposed method will not only reduce the workload of understanding and transforming the geometry problems needed to be done manually before, but also strengthen the efficiency and intelligence of existing geometry problem solvers.

The geometry problems solved in this paper are mainly the constructive geometry proving problems at secondary school level with no geometric quantities. The geometric elements in these problems are successively introduced and the geometric relations are relatively explicit. However, the proposed method can be also extended to understand the constructive geometry problems with geometric quantities, but owing to the limitation of existing geometry systems to tackle quantities [18, 20], it is difficult to process the transformed logical expressions containing quantitative relations. Presently, the proposed method cannot be directly used to understand the declarative geometry problems like "*The feet of the perpendiculars from a point to the sides of a triangle are collinear if and only if the point lies on the circumcircle of the triangle*". Problems of this type do not explicitly contain geometric relations. For these problems to be tackled, some processes are needed to transform them into the constructive forms and this is one of our future work.

In the future, the research in this paper can be also developed in the following three directions. Firstly, we plan to develop the improved automatic solver of plane geometry problems based on the method of understanding problems. Secondly, as various heuristics are used in the rule-based method in problem understanding, in the future, we want to use machine learning method to automatically learn the relation models from large numbers of exercise problems and use the learned models to perform relation extraction in new problems. Thirdly, we will also explore to fuse the propositions extracted from text in the question stem and the propositions extracted from diagrams to understand and formalize more kinds of exercise problems in plane geometry.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61272206) and the self-determined research funds of CCNU from the colleges' basic research and operation of MOE (No. ccnu16JYKX005). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. Chen, Representation and automated transformation of geometric statements, *Journal of Systems Science and Complexity*, vol.27, no.2, pp.382-412, 2014.

[2] W. K. Wong, S. C. Hsu, S. H. Wu et al., LIM-G: Learner-initiating instruction model based on cognitive knowledge for geometry word problem comprehension, *Computers & Education*, vol.48, no.4, pp.582-601, 2007.

[3] A. Mukherjee and U. Garain, A review of methods for automatic understanding of natural language mathematical problems, *Artificial Intelligence Review*, vol.29, no.2, pp.93-122, 2008.

[4] M. Seo, H. Hajishirzi, A. Farhadi et al., Solving geometry problems: Combining text and diagram interpretation, *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.17-21, 2015.

[5] A. Mukherjee and U. Garain, Understanding of natural language text for diagram drawing, *Proc. of the 13th International Conference on Artificial Intelligence and Soft Computing*, 2009.

[6] A. Mukherjee, U. Garain and M. Nasipuri, On construction of a GeometryNet, *Proc. of IASTED International Conference on Artificial Intelligence and Applications (AIA 2007)*, Innsbruck, Austria, pp.530-536, 2007.

[7] Q. T. Liu, H. Huang and L. J. Wu, Using restricted natural language for geometric construction, *Applied Mechanics and Materials*, pp.465-469, 2012.

[8] W. K. Wong, S. K. Yin and C. Z. Yang, Drawing dynamic geometry figures online with natural language for junior high school geometry, *International Review of Research in Open & Distance Learning*, vol.13, no.5, pp.126-147, 2012.

[9] H. Y. Guo, Q. T. Liu et al., Research for facing the natural language of the geometry drawing, *Computer Science*, vol.39, no.6A, pp.503-506, 2012.

[10] N. Do, H. P. Truong and T. T. Tran, An approach for translating mathematics problems in natural language to specification language COKB of intelligent education software, *IEEE International Conference on Artificial Intelligence and Education (ICAIE)*, pp.324-330, 2010.

[11] W. Lu, H. T. Ng, W. S. Lee et al., A generative model for parsing natural language to meaning representations, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.783-792, 2008.

[12] P. Clark and O. Etzioni, My computer is an honor student – But how intelligent is it? Standardized tests as a measure of AI, *AI Magazine*, vol.37, no.1, pp.5-12, 2016.

[13] Z. Ye, S. C. Chou and X. S. Gao, An introduction to java geometry expert, *International Workshop on Automated Deduction in Geometry*, Springer Berlin Heidelberg, pp.189-195, 2008.

[14] J. Narboux, GeoProof: A user interface for formal proofs in geometry, *Journal of Automated Reasoning*, vol.39, no.2, pp.161-180, 2007.

[15] D. Wang, GEOTHER 1.1: Handling and proving geometric theorems automatically, *International Workshop on Automated Deduction in Geometry*, Springer Berlin Heidelberg, pp.194-215, 2002.

[16] G. J. Richter and U. H. Kortenkamp, The interactive geometry software Cinderella.2, *American Mathematical Monthly*, 1999.

[17] F. Botana, M. Hohenwarter, P. Janicic and S. Weitzhofer, Automated theorem proving in GeoGebra: Current achievements, *Journal of Automated Reasoning*, vol.55, no.1, pp.39-59, 2015.

[18] J. G. Jiang and J. Z. Zhang, A review and prospect of readable machine proofs for geometry theorems, *Journal of Systems Science and Complexity*, vol.25, no.4, pp.802-820, 2012.

[19] K. Wang and Z. Su, Automated geometry theorem proving for human-readable proofs, *Proc. of the 24th International Conference on Artificial Intelligence (AAAI)*, pp.1193-1199, 2015.

[20] Z. Ye, S. C. Chou and X. S. Gao, Visually dynamic presentation of proofs in plane geometry, *Journal of Automated Reasoning*, vol.45, no.3, pp.213-241, 2010.

[21] Y. Tian, *Everyday Practice of Plain Geometry Volume I: Foundation Part (Linear Type)*, Harbin Institute of Technology Press, 2013.

[22] S. C. Chou, *Mechanical Geometry Theorem Proving*, Springer Science & Business Media, 1988.

[23] H. P. Zhang, H. K. Yu, D. Y. Xiong et al., HHMM-based Chinese lexical analyzer ICTCLAS, *Proc. of the 2nd SIGHAN Workshop on Chinese Language Processing*, vol.17, pp.184-187, 2003.

[24] V. Aleven, I. Roll, B. M. McLaren et al., Help helps, but only so much: Research on help seeking with intelligent tutoring systems, *International Journal of Artificial Intelligence in Education*, vol.26, no.1, pp.205-223, 2016.

[25] M. Ganesalingam and W. T. Gowers, A fully automatic theorem prover with human-style output, *Journal of Automated Reasoning*, vol.58, no.2, pp.253-291, 2017.

[26] B. Cabaleiro, A. Peas and S. Manandhar, Grounding proposition stores for question answering over linked data, *Knowledge-Based Systems*, vol.128, pp.34-42, 2017.

[27] P. Boutry, G. Braun and J. Narboux, Formalization of the arithmetization of Euclidean plane geometry and applications, *Journal of Symbolic Computation*, pp.23-46, 2017.