# The Question Answering System of DCUMT in NTCIR-11 QA Lab

Tsuyoshi Okita
Dublin City University
tokita@computing.dcu.ie

Qun Liu
Dublin City University
qliu@computing.dcu.ie

## ABSTRACT

This paper describes the question answering system developed at Dublin City University for participation in the QA Lab shared task in NTCIR-11 [20]. We participated in three tasks: center exam (multiple choice) tasks in Phases 1 and 2, and secondary exam task (written) in Phase 2. We built a QA system in which we use the specialized-purpose parser *KBarse* to acquire meaning representation which is called case frame graphs from history textbooks using commonsense knowledge. We used distributed representation for Out-Of-Vocaburary (OOV) words and missing assertions. We added prototype functionality for handling implicit arguments/relations, causality analysis, time analysis, and temporal order analysis by heuristics. Our results for center exam task in Phase 1 was 77.0 which was the first among seven submissions, for center exam task in Phase 2 was 72.0 which was the first among nine submissions, and for secondary exam task in Phase 2 in terms of precision were 71.4 (UTokyo), 62.5 (KyotoU), 71.8 (Hokkaido), 62.7 (Waseda), and 80.0 (Chuo) which were the first among two submissions.

## Team Name

DCUMT

## Subtasks

Center Exams (Phase 1 and 2), Secondary Exam (Phase 2)

## Keywords

Question Answering, Semantic Parser, knowledge base

## 1. INTRODUCTION

This paper describes the question answering system developed at Dublin City University for participation in the QA Lab shared task in NTCIR-11 [20]. We participated in three tasks: center exam (multiple choice) tasks in Phases 1 and 2, and secondary exam (written) task in Phase 2.

We built a Question Answering (QA) system in which we use the specialized-purpose parser *KBarse* to acquire meaning representation which is called case frame graphs from history textbooks using commonsense knowledge. We used distributed representation. We added prototype functionality for handling implicit arguments/relations, causality analysis, time analysis, and temporal order analysis by heuristics.

The remainder of this paper is organized as follows. Section 2 describes the overview of our systems. Our experimental results are presented in Section3. We conclude in Section 4.

## 2. OUR SYSTEMS

Semantic parsing is the process of mapping a sentence into a formal representation of its meaning [6, 13, 2, 12]. There are two kinds depending on the capability of automated reasoning which is useful for QA task: (1) a case-role analysis without automated reasoning, and (2) a deeper semantic analysis attached with the predicate logic or other formal language with automated reasoning. Since we use both of these in this paper we call the former the type-I semantic parser and the latter the type-II semantic parser. The former category of Japansese semantic parser, i.e. type-I semantic parser, is provided by KNP [9, 10]. KNP provides an analysis of the case structure, which corresponds to the Japanese specific predicate-argument relations, as semantic representation but does not provide predicate logic. One characteristic of Japanese is in the function of the Japanese particles, which are called *jyoshi*, is to indicate various meanings and functions such as speaker affect and assertiveness by following the modified noun, verb, adjective, or sentence. Such case structure can be reordered often without changing their meanings which contrast Japanese with other languages: by this phenomenon Japanese is categorized as a free-order langauge. Due to this characteristic, the case structure can be thought of as one appropriate level of abstraction to capture semantic representation in the case of Japanese. Thus, we use the case structure as the basic semantic representation in this paper. The latter category of semantic parser, i.e. type-II semantic parser, can be used in the context of QA task. The resulted semantic representation with predicate logic can be specifically used to evaluate itself with the knowledge base: in the first step a question is converted into the equivalent semantic representation with predicate logic, and in the second step this semantic representation is evaluated with knowledge base to yield an answer.

This paper takes the approach to use the former category of semantic parser *KBarse* to add the functionality of predicate logic. That is, in terms of obtaining the semantic representation we rely wholy on *KBarse*. Then, we added predicate logic to them. In the construction of knowledge base, we use *KBarse* to obtain a set of predicate-argument relations (or case frames). In the question preparation, we use *KBarse* to obtain a set of predicate-argument relations (or case frames) and add predicate logic.

## 2.1 Overview as QA Systems

Let $x$ be utterance (*question*), $y$ be the *answer*, $z$ be a logical form of $x$, and $k$ be knowledge base.

Definition 1 (Question Answering-Type Method). *A QA task can be divided into three subtasks: knowledge base construction subtask, question preparation subtask (via type-II semantic parsing $z \sim p(z|x;\theta)$), and question evaluation subtask (via semantic evaluation $y = [\![z]\!]_k$). (Readers who want to know the details related to these should read [13, 12].)*

In order to give explanation to the knowledge base construction, let $E$ be a set of entities (e.g., アルハンブラ宮殿) and let $P$ denote a set of properties (e.g., 建てる$_{\text{動}1}$). A knowledge base $K$ is a set of assertions $(p, e_1, e_2)$ ($\in P \times E \times E$) or $(p, e_1, e_2, \ldots)$ ($\in P \times E \times E \times \ldots$)) (e.g., 建てる$_{\text{動}1}$ ナスル朝$_{\text{ガ}}$ グラナダ$_{\text{ニ}}$ アルハンブラ宮殿$_{\text{ヲ}}$). (Similarly, a knowledge base $K'$ can be seen as a graph/tree of assertions.) We have 14k of properties consisting of Yamakawa and Tokyo Shoseki textbooks whose assertions are 80k. (Freebase of Google has 19k properties but 596M assertions).

A type-II semantic parser maps new questions $x$ to answers $y$. A type-II semantic parser is obtained by solving the mapping of new questions $x$ to answers $y$ for given knowledge base $K$ and a training set of question-answer pairs. It is noted that in practice we obtain the semantic representation of new question $x$ by *KBarse* (type-I semantic parser) and add predicate logic to this. First, we assume that historical textbook will not contain inconsistency. Second, we assume that the question will not be related to the voting type judgement. Third, we assume that the question will be limited to a simple one: for example, if we need to detect a yes-no judgement, we assume that one of their questions are at least explicit. Under these assumptions we did not deploy a function of learning but only a function of logics in semantic parser.

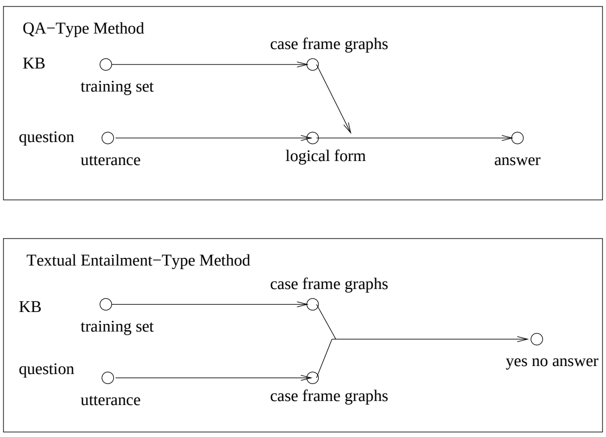


**Figure 1: Overview of two kinds of methods.**

We consider two kinds of basic solutions which are depicted in Figure 1. Suppose we have a knowledge base $K$ which is a set of assertions, which can be seen as a database.

| | |
|---|---|
| Utterance | ナスル朝が，グラナダにアルハンブラ宮殿を建てた。 |
| Assertions | 建てる$_{\text{動}1}$ ナスル朝$_{\text{ガ}}$ グラナダ$_{\text{ニ}}$ アルハンブラ宮殿$_{\text{ヲ}}$ |
| Typed-Assertions | 建てる$_{\text{動}1}$ ナスル朝$_{\text{era}}$ グラナダ$_{\text{location}}$ アルハンブラ宮殿$_{\text{building}}$ |
| Utterance | アルハンブラ宮殿は何朝が建てられたのか。 |
| Assertions | 建てる$_{\text{動}1}$ とき$_{\text{ガ}}$ の$_{\text{外の関係}}$ アルハンブラ宮殿$_{\text{ヲ}}$ |
| Logical form (lambda calculus) | $\lambda x.\exists e$ 建てる $(x_{\text{朝}}, e) \bigwedge$ building(e, アルハンブラ宮殿) |
| Entities | ナスル朝, グラナダ, アルハンブラ宮殿 |
| Property | 建てる$_{\text{動}1}$ |
| Type of entities | era=type(ナスル朝)<br>location=type(グラナダ)<br>building=type(アルハンブラ宮殿) |

**Table 1: Terminology used in this paper.**

A general form of simple QA task is thus similar to the database search and to pose a query to this knowledge base $K$ [13, 12, 2]. We need to convert a question posed in natural language into a logical form. Then, this logical form is evaluated on $K$, which yields an answer.

A multiple choice question can be a special kind of QA tasks. In this case, we will match a graph in question and a graph in knowledge base $K$, which can be evaluated by the graph-based matching [8, 15].

Definition 2 (Textual Entailment-Type Method). *Let us call a Hypothesis $H$ and a Text $T$ (The former comes from a question while the latter is a knowledge base $K$.). We parse both of $H$ and $T$ to obtain assertions. If a graph in $H$ and in $T$ matches, we call $H$ is true, otherwise false.*

## 2.2 Knowledge Base Construction Step: Type-I Semantic Parsing

As is mentioned at the beginning of this section, we deploy the type-I semantic parser to obtain the semantic representation. Since we modified various specification as is mentioned in this section we call it *KBarse* although our semantic parser bases on KNP.

In the initial phase we conducted an experiment to measure the performance of KNP [10] where we only used the case frame extractor, how much predicate-argument relations it extracts from the original text. It is noted that the predicate-argument relations are interchangeably expressed as the case frames below which are identical in our context.

Definition 3 (Relation Coverage Ratio (RCR)). *Prepare the extracted predicate-argument relations from the original text and discard all the functional words such as jyoshi and jyodoushi from the original text. Then, we define the relation coverage ratio as the ratio of the number of words extracted as predicate-argument relations divided by the number of words in original text. It is noted that words can be replaced by morpheme which are often appropriate level to measure this ration in the case of Japanese.*

We compared only the number of content words, discarding all the functional words such as jyoshi and jyodoushi. The Relation Coverage Ratio (RCR), which we measure at the morpheme level, was 57.2%. The ratio RCR indicates that around 42.8% of possibly morpheme, which may be related to the important words for some questions, in the original text are disapeared. Our analysis of the important drops of words/phrases were related to the following expressions. First, the coordinated words/phrases and the parallel expressions are often problematic and were not extracted correctly. Second, if we focus on the dropped words/phrases they were often related to the noun predicates. On compared to this the verb predicates were fairly good and robust. Third, although this may be related to the specification of predicate-argument relations, "is-a relations", which are one of the major actors in the relation extraction literature [17] or Google knowledge graphs, were often dropped in the noun predicates.

The predicate-argment relations indicate the pairwise one-to-many relationship between the verb/noun predicates and their arguments as well as their modifiers. For the sake of the construction of the knowledge base which is useful for QA tasks, it is better that we have higher RCR for the knowledge base, which is nearly as 100%. That is, we do not want to let some words/phrases in predicate-argument relations escaped from the beginning. This is since it is true that many of the elements in a text has texture which are in some way related to other elements, which may be lost. In this reason, we modified several grammatical specification of the predicate-argument relations. The following specification shows the major differences of KNP and KBarse as well as the new features of KBarse such as temporal/causal relation analyzers and topic stamps.

Example 1 (Verb Predicate). *The following example shows the parsing results.*

| |
|---|
| 戦国時代に西の辺境にあった秦は，前 *221* 年にはじめて統一をなしとげた。 |
| *(KNP/KBarse)* 名 *0:3* 辺境 西 ノ<br>*(KNP/KBarse)* 動 *7:3* 有る 秦 ガ 戦国時代 時間 辺境 ニ<br>*(KNP/KBarse)* 動 *6:3* 始める 前 *221* 年 時間 秦 ガ<br>*(KNP/KBarse)* 動 *1:3* 成す%遂げる 秦 ガ 統一 ヲ |

Example 2 (isA relation). *It is often the case when the sentence is ended with* である*, the predicate is a noun predicate. In this case the semantic parsing may not capture the subject for this noun predicate. This is often related to the case when this subject and the noun predicate forms a "isA relation".*

| |
|---|
| クレオパトラは女王である。 |
| *(KNP)* 名 *1:2* 女王 |
| *(KBarse)* 動 *1:3 isA* クレオパトラ ハ 女王 |

Example 3 (Noun Predicate). *When the predicate is a noun predicate of* 息子*, this interpretation will drop an important isA relation between* アレクサンドロス *and* フィリッポス *2* 世の息子*. Importantly,* アレクサンドロス *will be disapeared from the case frames.*

| |
|---|
| アレクサンドロスはフィリッポス *2* 世の息子である。 |
| *(KNP)* 名 *1:3* 息子 フィリッポス *2* 世 ノ |
| *(KBarse)* 名 *1:3* 息子 フィリッポス *2* 世 ノ<br>*(KBarse) isA:2* アレクサンドロス ハ フィリッポス *2* 世の息子 |

Example 4 (Coordination). *The former elements in coordination is often dropped. That is among two elements* カラコルム *and* 大都*, the first one* カラコルム *is dropped.*

| |
|---|
| 同じころ，ヨーロッパからの使節や商人などもカラコルムや大都をおとずれた。 |
| *(KNP)* 名 *1:2* 使節 *&*<br>*(KNP)* 名 *1:2* 商人 *&*<br>*(KNP)* 動 *1:3* 訪れる 同じ 修飾 商人 ガ 大都 ヲ |
| *(KBarse)* 動 *1:3* 訪れる 同じ 修飾 使節や商人 ガ カラコルムや大都 ヲ |

Example 5 (Parallel expression). *When the parenthesis is used such as "A（B）", A is dropped. That is, in the case below,* 『コーラン』（『クルアーン』）『コーラン』 *is dropped.*

| |
|---|
| アッラーの啓示がイスラームの聖典『コーラン』（『クルアーン』）である。 |
| *(KNP)* 名 *1:3* 聖典 イスラーム ノ クルアーン ノ |
| *(KBarse)* 名 *1:3* 聖典 イスラーム ノ 『コーラン』（『クルアーン』） ノ |

### *Temporal/Causal Relationships.*

One idea to capture the causal and temporal relationships is the parser based on the Rhetorical Structure Theory (RST) [18]. The necessity of such parser comes from the fact that the coverage of causal and temporal relationships in the manually created resources such as FrameNet [1], WordNet [5], which encode many aspects of commonsense knowledge, remains low for many domains [7].

Example 6 (Temporal Relationship). *There is a temporal order in two events [event1* 唐の滅亡後 *] and [event2* 華北では五つの王朝が交替し *] in this order. Similarly, there is another order betweeen [event1* 唐の滅亡後 *] and [event3* 長江流域などの地方でも *10* の国が興亡した（五代十国）*].*

| |
|---|
| 唐の滅亡後，華北では五つの王朝が交替し，長江流域などの地方でも *10* の国が興亡した（五代十国）。 |
| [*before* 唐の滅亡後 ]，[*after* 華北では五つの王朝が交替し ]，<br>[*before* 唐の滅亡後 ]，[*after* 長江流域などの地方でも *10* の国が興亡した（五代十国）]。 |

Example 7 (Causal Relationship). *There is a cause-effect relationship between two events [cause* 唐末以後の混乱は *] [effect* 周辺諸国の興亡や諸民族の自立をうながした *].*

| |
|---|
| 唐末以後の混乱は，周辺諸国の興亡や諸民族の自立をうながした。 |
| [*cause* 唐末以後の混乱は ]，[*effect* 周辺諸国の興亡や諸民族の自立をうながした ]。 |

### *Topic Stamps.*

We intend to check whether entities whose time/location stamps are consistent or not among multiple of assertions if they needs to combine together. Similarly, we can use time/location stames to check whether multiple entities that are contained in the question are consistent in terms of time/location or not.

Example 8 (Time Stamp). *We label the topic stamp for each assertion (predicate-argument relation) according as their scope. We assume that although there might be multiple time events in a sentence each assertion does not include more than two time events. Since only handful of assertions include time events, they are propagated to other assertions.*

| |
|---|
| *12* 世紀にカスティリャから独立したポルトガルでは，*15* 世紀後半，国王ジョアン２世が王権を強化した。 |
| 動 *3:3* 独立 ポルトガル $_{\text{ガ}}$ *12* 世紀 $_{\text{時間}}$ カスティリャカラ $<$ *time* :*12* 世紀 $>$<br>名 *1:3* 王権 ポルトガル $_{\text{ノ}}$<br>動 *1:3* 強化 ポルトガル $_{\text{デ}}$ *15* 世紀後半 $_{\text{時間}}$ 国王ジョアン２世 $_{\text{ガ}}$ 王権 $_{\text{ヲ}}$ $<$ *time* :*15* 世紀後半 $>$ |

Example 9 (Location Stamp). *It is often preceded by* では *(e.g.* イギリスでは*), which are often represented as "de"-case (e.g.* デ格*).*

| |
|---|
| イベリア半島では *10*〜*11* 世紀にカスティリャ，アラゴンの両王国が成立した。 |
| 動 *1:3* 成立 イベリア半島 $_{\text{デ}}$ *11* 世紀 $_{\text{時間}}$ 両王国 $_{\text{ガ}}$<br>$<$ *loc* : イベリア半島 $>$ |

| |
|---|
| イギリスでは，その後ばら戦争とよばれる内戦がおこり，そのあいだに多くの諸侯や騎士は没落した。 |
| 動 *5:3* 呼ぶ%れる 内戦 $_{\text{ガ}}$ ばら戦争 $_{\text{ト}}$<br>名 *1:3* 内戦 イギリス $_{\text{ノ}}$<br>動 *3:3* 興る その後 $_{\text{時間}}$ 内戦 $_{\text{ガ}}$ イギリス $_{\text{デ}}$ $<$ *loc* : イギリス $>$<br>名 *1:2* 多く *&*<br>名 *1:3* 諸侯 多く $_{\text{ノ}}$<br>名 *1:3* 騎士 多く $_{\text{ノ}}$ イギリス $_{\text{ノ}}$<br>動 *3:3* 没落 あいだ $_{\text{時間}}$ 騎士 $_{\text{ガ}}$ |

### *Paraphrasing.*

It is often that country and person co-appeared.

Example 10 (Cooccurence). アッバース朝*,* 第 *5* 代カリフ*, and* ハールーン＝アッラシード *are cooccured.*

| |
|---|
| アッバース朝は，第 *5* 代カリフのハールーン＝アッラシードの時代に最盛期を迎えた。江南にのがれた徽宗の子の高宗 |

### *Coordination.*

Investigation of coordination will make the prediction of unseen word, at least the type of the word.

Example 11 (Coordination). *In the following example, we can consider* ハディース *similar with*『コーラン』*.*

| |
|---|
| 『コーラン』もハディースも，アラビア語で書かれているため，アラビア語がイスラーム世界の共通語となっている |

| | |
|---|---|
| type of entity | 建物, 宮殿, 建造物, 何宮殿, 建築物, 何という名前の宮殿,... |
| entity | アルハンブラ宮殿 |

**Table 2: Table shows the types which are possibly embedded in training set of questions.**

### *Others.*

There are many sentences which do not yield case frames for some fragments in a sentence. Even in such a case, if we change the structure of the problematic fragments, such as voice, predicate, etc, it often worked.

## 2.3 Knowledge Base Construction Step: Entities and Their Types

Entity types are important items for the question evaluation subtask. Suppose that we have a question "アルハンブラ宮殿は何朝に建てられたのか", that is "$\lambda x.\exists e$ 建てる ($x_{\text{朝}}$,e)$\wedge$building(e, アルハンブラ宮殿)". In this case, We need to seek entity $e$ which can be simultaneously match two predicate-argument relations "建てる ($x_{\text{朝}}$,e)" and "building(e, アルハンブラ宮殿)". Let our knowledge base consisting of only one typed-assertion "建てる $_{\text{動}1}$ ナスル朝 $_{\text{era}}$ グラナダ $_{\text{location}}$ アルハンブラ宮殿 $_{\text{building}}$". We start from the first statement 建てる ($x_{\text{朝}}$,$e_*$). Athough we can use any entity $e_*$ as a connection with the second term in a question but since we have only one item in knowledge base, we start with the logic "建てる ($x_{\text{朝}}$, グラナダ $_{\text{location}}$; x=ナスル朝)", From the knowledge base, we also have "building(グラナダ $_{\text{location}}$, アルハンブラ宮殿)" which validates the statement. Hence, an answer in this case is produced as"ナスル朝". This example show that the entity $e_*$ which works as a connection between the series of terms although it did not provide more than two options in this example. In this way we can say that this entity and its type are key element in this framework.

Once we define such a procedure, we can use this entity to control the quality of the match. That is to use the variable type definition for the entities depending on the case. For example, we define the type of location, consisting of three entities "グラナダ", "スペイン", and "イベリア半島". However, in another case, we define three different type of location consisting of "グラナダ" , type of nation consisting of "スペイン", and type of peninsula consisting of "イベリア半島". Using these different diffinition of type, we can control the match by entity. As we mention in this subsection, we define the notion of type step by step.

The major source of the definition of type comes from the following three.

- Examine question-answering pair of the training set.
- Examine the ranked list of single OOV words. In the case of morphological analyzer JUMAN [11], these are acquired automatically from Wikipedia with their categories. (We extracted entity types looking up the Wiki for the possible abstract names. See Table 3)
- Examine the frequency of words in the whole documents (after morphological analysis). From the top, we guess which words are likely to be among the object of question-answering pair, and then we consider the possibility of compound words which can be grow

in the left side in the case of Japanese. For example, 朝 can be a good source for compound words if it is combined in its left side, say ナスル. (See Table 4)

If these conditions met, we consider them as the registered words. As in the case of RCR, we define the entity coverage ratio.

DEFINITION 4 (ENTITY COVERAGE RATIO (ECR)). *Prepare the knowledge base consisting of nontyped-assertions and count the number of all the entities in assertions (=A). Obtain the registered entity type by three methods just mentioned above and count the number of all the entities whose type are defined (=B). Then, we define the entity coverage ratio as the ratio of $A/B$.*

Since the number of definition of type are as much as the number of entities which is fairly big, this process to allocate the types for entities takes time and efforts. In sum, in relation with this type-entity relationship, there is a trade-off in quality of QA systems. Hence, on the one hand, if the question evaluation (or inference) yields more than two answers, we can control it by reducing the number of types for the entity cluster. On the other hand, if the question evaluation does not yield any answer (when we predict that we obtain some answer), we can control it by making the number of types for the entity clusters small.

## 2.4 Question Evaluation Step

For the given question, we first seek the answer whether one assertion contains both of these. If one assertion contains both of question and answer, it does not need to find over two assertions but we obtain the answer. Otherwise, it will let increase the length of the number of assertions as much as 1: that is, we combine two assertions and we seek the type of answer in the second assertions.

As is noted in the previous subsection, the granularity of type often matters the time/quality to obtain the answer. We change the definition of granularity adaptively, making some type containing a lot of or very small number of entities.

Noted the similarity of letting increase the length of the number of assertions and latent variable models.

However, it is often necessary that we poke more than two assertions.

### *Commonsense Knowledge: Heuristics.*

There are several possible situation which requires commonsense knowledge.

The first situation is related to the question. For example, suppose that the question is related to numbers of persons, i.e. 「18 世紀に日本に来た西洋人は何ヵ国以上いたと考えられるか？」 In this case, we will need to employ commonsense knowledge to formulate a different question than the original one. That is, we need to pose a question whether 18 世紀にイギリス人は日本へ来たか and 18 世紀にアメリカ人は日本へ来たか. the answer would be 二ヵ国以上. It is generally known that such commonsense knowledge/world knowledge are necessary for QA/textual entailment tasks. In this case, we used heuristics to modify the question in logical form. It is noted that this would only cover a handful of questions. It is noted that without changing the question the evaluation stage can handle this but seems more complex.

Similarly, there are various situations which are related to the knowledge base. The case of knowledge base is related to

| | | |
|---|---|---|
| 都市 | 59 | アガディール, アストラハン, アゾフ, アフラ, アントウェルペン, イスタンブル, イズミル,... |
| 男性名 | 46 | アフマド, アルダシール, アレクサンドル, アンリ, イドリース, イムレ, ... |
| 姓 | 31 | アントニウス, エーベルト, オットー, カペー, クレマンソー, グエン, ゲーリング, ... |
| 称号 | 16 | アミール, カウディーリョ, カリフ, サトラップ, シャー, ツァーリ, ナイト, ニザーム, ... |
| 州都 | 14 | カーブル, コルカタ, サンクトペテルブルク, チェンナイ, ノヴゴロド, ハイデラバード, ... |
| 元号 | 13 | 安政, 嘉隆, 開元, 康熙, 弘安, 洪武, 崇禎, 宣統, 宣徳, 貞享, 文永, 咸豊, 雍正 |
| 人名 | 11 | アンティゴノス, カルロス, グスタフ, ダレイオス, テオドシウス, バティスタ,... |
| 県庁所在地 | 11 | アミアン, アレクサンドリア, アヴィニョン, オルレアン, シャルトル, スエズ, ニース, ... |
| 呼称 | 10 | アンダルス, インディアン, ガリア, クエーカー, シルクロード, バルバロイ, 蝦夷, ... |
| 地名 | 10 | アカディア, オコンネル, カニング, クスコ, クレルモン, ゲルマニア, ディアス, ... |
| 地域 | 10 | カフカース, スカンディナヴィア, チロル, トルキスタン, パンジャーブ, フランドル,... |
| コムーネ | 8 | アッシジ, アナーニ, カッシーノ, グリマルディ, サッコ, ラヴェンナ, ヴィンチ, ヴェネツィア |
| 基礎自治体 | 8 | カノッサ, コリントス, デルフト, トリエステ, マーストリヒト, ユトレヒト, ロカルノ,... |
| 女性名 | 8 | イサベル, エリザベス, カトリーヌ, キャサリン, ジャンヌ, テレジア, ヘレン, ... |
| 人々 | 7 | クリオーリョ, スーフィー, ヒクソス, ヒッピー, マオリ, マワーリー, メスティーソ |
| 哲学者 | 7 | アリストテレス, エピクテトス, エピクロス, タレス, デモクリトス, ピタゴラス, プラトン |
| 港湾都市 | 7 | アデン, カリカット, スーラト, バスラ, ブリストル, プリマス, ホルムズ |
| 王朝 | 7 | 胡朝, 丁朝, 陳朝, 南宋, 北宋, 阮朝, 黎朝 |
| 首都 | 6 | カザン, コロンボ, テノチティトラン, ブレーメン, モガディシュ, 平城京 |
| 遊牧国家 | 5 | エフタル, スキタイ, ハザール, 柔然, 匈奴 |
| 地方 | 5 | アルスター, アンジュー, イオニア, ヒジャーズ, ラインラント |
| 一柱 | 5 | オシリス, ガネーシャ, シヴァ, フォラス, 梵天 |
| 古名 | 5 | アテナイ, ソグディアナ, バクトリア, ヒスパニア, ペルシア |
| 王国 | 5 | アステカ, バビロニア, パルティア, ミタンニ, リディア |
| 干支 | 5 | 乙丑, 壬午, 壬辰, 丁酉, 戊戌 |
| 町 | 5 | オホーツク, ティエール, トルデシリャス, ヒヴァ, ミレトス |
| 行為 | 4 | 侮蔑, 焚書, 放伐, 沐浴 |
| 一派 | 4 | ボリシェヴィキ, 全真教, 南宗, 北宗 |
| 島 | 4 | グリーンランド, サルデーニャ, トゥーレ, 樺太 |
| 遺跡 | 4 | クテシフォン, クノッソス, テーベ, ティリンス |

**Table 3: Table shows the ranked list from single OOV words.**

| | | |
|---|---|---|
| 年 | 3549 | 1500 年,1501 年,1505 年,1506 年,1510 年,1511 年,1513 年,1517 年,1519 年,1520 年,1521 年, |
| 世紀 | 2144 | 2 世紀, 3 世紀, 4 世紀, 5 世紀, 6 世紀, 7 世紀, 8 世紀, 9 世紀, 10 世紀, 11 世紀, 12 世紀, |
| 人 | 1712 | アイルランド人, アステカ人, アッカド人, アッシリア人, アテネ人, アフリカ人, アムル人, |
| 国 | 1631 | アウド王国, アクスム王国, アステカ王国, アチェー王国, アッシリア王国, アユタヤ王国, |
| 世界 | 1305 | イスラーム世界, ヘレニズム世界, 地中海世界, 第三世界, オリエント世界, 戦後世界, 海域世界, |
| 帝国 | 1053 | ビザンツ帝国, モンゴル帝国, 西ローマ帝国, ペルシア帝国, アステカ帝国, アッシリア帝国, |
| 戦争 | 1025 | ばら戦争, アウクスブルク同盟戦争, アジア太平洋戦争, アフガニスタン戦争, アフガン戦争, |
| 者 | 1024 | 労働者, 指導者, 聖職者, 支配者, 後継者, 失業者, 社会主義者, 保護者, 手工業者, 独裁者, 擁護者, |
| 社会 | 994 | 封建社会, 地域社会, インド社会, アメリカ社会, ポリス社会, 遊牧社会, 近代社会, 複合社会, |
| 化 | 970 | 工業化, 近代化, イスラーム化, 民主化, 一体化, 国有化, 中央集権化, 機械化, 長期化, トルコ化, |
| 主義 | 922 | コルベール主義, カルヴァン主義, 孤立主義, 小ドイツ主義, 文治主義, 徳治主義, 修正主義, |
| 経済 | 918 | 貨幣経済, 市場経済, 都市経済, ブロック経済, 主義経済, 現物経済, 統制経済, バブル経済, |
| 支配 | 897 | 専制支配, 皇帝支配, 領土支配, 帝国支配, セレウコス朝支配, 強権支配, 寡頭支配, |
| 朝 | 876 | アイユーブ朝, アケメネス朝, アッバース朝, アユタヤ朝, アリー朝, アルサケス朝, アンコール朝, |
| 運動 | 833 | キュロット運動, 洋務運動, 五・四運動, バクティ運動, ラダイト運動, ブルシェンシャフト運動, |
| 地 | 805 | 植民地, 公有地, 居留地, ケープ植民地, 要地, 中継地, 供給地, 領主直営地, 租借地, 故地, |
| 革命 | 794 | 二月革命, 三月革命, 七月革命, 十月革命, 価格革命, 農業革命, 商業革命, 交通革命, |
| 民族 | 765 | ゲルマン民族, スラヴ民族, セム系諸民族, ケルト系民族, 少数民族, 異民族, チベット民族, |
| 軍 | 758 | アフガン軍, アメリカ軍, アラブ軍, アルビジョワ十字軍, イェニチェリ軍団, イギリス軍, |
| 独立 | 726 | 形式的独立, 共和国独立, ポーランド独立, ベルギー独立, ハイティ独立, オランダ独立, |

**Table 4: Table shows the frequency of words in the whole documents.**

give them the correct type of entities. For example, suppose that some entities include various expressions such as 1039 年, 39 年 3 月, 1034 年〜43 年, or 1039 年 3 月 9 日朝. It may need that all of these date should be handled in the same manner in their types which depend on the question.

- (number counting) 1 世紀たらずで 4 王国に分裂した。 Even though we may know the name of these four countries, there is no connection between these knowledge and the fact that these are four countries.
- (time ordering) The temporal order of 100 年 and 前 100 年 are 前 100 年 < 100 年.
- (valid time range) The description about 2000 年 is not a reality.
- (multiple times) Some country adopts the system of sub-calendar (e.g. 年号).
- (locational overlapping) 東アジア諸国 consists of several countries.
- (indication of multiple combination) 東西フランク王国 consists of 東フランク王国 and 西フランク王国.
- (daily life) 映画王国 and 天国 are not among countries.
- (naming convention) ヴァスコ＝ダ＝ガマ. Some may say by name ガマ or ヴァスコ, by titles, or by other names.
- (complex question) 航海に役立つ器具を発明したのはどこの国でのことか。

*Commonsense Knowledge: Distributed Representation.*

Distributed representation can be used for the sake of zero-shot learning [16, 14, 15]. Although we used the type-I semantic parser which is not built by deep learning, it is possible for us to built the distributed representation in order to aim at capturing Out-Of-Vocaburary (OOV) words and missing assertions. The latter suggests the lack of some description of entities (This may be related to anaphora) or missing assertions completely. First, this situation is probable since the question does not need to be written in the same words which are not appeared in the training set. Both OOV words and missing assertions are applied. Second, by commonsense knowledge we can think that this is reasonable since we can interpret them as the same kinds with OOV words and missing assertions.

We construct the phrase-level word embeddings by word2vec [14]. Then, considering the recursive way of constructing word embeddings [21, 22, 3], we use RNN encoder-decoder to map the predicate-level word embeddings, and the following predicate-sequence-level word embeddings. In this construction, on the basis of the phrase-level word embeddings, the two following word embeddings are additionally embeded on the same real-value vectored space.

## 2.5 Experiments

*Question Format Types.*

Up until now, we have described mostly the QA-type method and very small mentioning about the textual entailment-type method in the subsection in 2.1. These two are the fundamental mechanism to solve the question.

In practice, the system requires to convert varieties of question type into the format of these methods. For example center exams will require to build the system which can handel at least (1) Yes No, (2) chronological ordering, (3) slot filling, (4) combination, (5) need-figures, (6) choose correct choice or wrong choise, and so forth. Hence, we set up in this way to convert these question type into the one which fits into our two fundamental methods. Similarly, we set up the question types for secondary exams automatically by the answer types which are mentioned in the next paragraph. Our system does not read images and does not properly handle to the type of "explain by natural language".

*Answer Types.*

We believe that the answer types depend on the underlying text. In this case, the texts are history textbook used for Japanese highschool students. We used the same types which we mentioned for the types for entities, that is person, organization, location, time, numerics, reason, relation, etc. In this reason, the categorization is variable which depends on the question.

### *Context in Query.*

It is often the case that the query side provides the context. Such context is often converted into the logical form using the conjunction and disjunction operators. When the query side consists of multiple sentences, our methods will generate the list of case frame graphs and converted into the logical form.

### *Query Expansion/Pseudo/Recursive Relevance Feedback.*

Although our system does not do query expansion, these are not directly related to our experiments. However, indeed there is alternative way to interprete our QA system which is in the same perspectives with C-DSSM [19] or in some part similar with OpenIE[4]. In this case, our system does the query expansion using the type-I semantic parser, doing the parsing of the answer side as well to obtain the case frame graph for the answer side. Based on this case frame graph, we did the query expansion. For the query expansion, we prioritize their queries case by case on the targeted case frame graph.

Similarly although our system does not do query expansion, if we do the alternative interpretation our system does a pseudo relevance feedback. When appropriate, we do the relevance feedback successively for the unattained words/phrases for the feedback.

### *Experimental Results.*

We used the training data of textbook data of Yamakawa shuppan and Tokyo Shoseki, which are provided by the QA-Lab organizers, and Wikipedia. The statistics of these training data and our knowledge base are shown in Table 5.

| History textbook | statistics (sentences) | |
|---|---|---|
| Yamakawa | | 5,482 |
| Tokyo Shoseki | | 12,448 |
| total | | 17,930 |
| | KNP | KBarse |
| case frames | 66,365 | 110,772 |
| location | 3,693 | 6,834 |
| person | 852 | 1,654 |

**Table 5: Statistics of history textbooks and knowledge base.**

Our results are shown in Table 6. (Refer to [20] for the other details of each test set.) Center exam for the phase 1 was the score of 77, for the phase 2 was the score of 72. In the phase 2, there were five kinds of exams. The precision of these five exams were 71.4 (UTokyo), 62.5 (KyotoU), 71.8 (Hokkaido), 62.7 (Waseda), and 80.0 (Chuo).

The RER was 98.3% and the ECR was 90.1% $\sim$ 95.2%. The distributed representation was equally effective for heuristics for the center exam in the phase 2, which is shown in Table 7.

| exam name | marks | correct | incorrect | NA |
|---|---|---|---|---|
| Phase 1 | | | | |
| Center Exam | 77.0/100.0 | 28 | 8 | |
| Phase 2 | | | | |
| Center Exam | 72.0/100.0 | 28 | 13 | |
| Secondary Exam | | | | |
| CU_Lit(247) | 32/40 | 32 | 8 | 14 |
| HU(844) | 23/32 | 23 | 9 | 10 |
| KU(750) | 30/48 | 30 | 18 | 5 |
| UT(792) | 5/7 | 5 | 2 | 6 |
| WU_Edu(476) | 27/43 | 27 | 16 | 7 |

**Table 6: Results for 7 kinds of test sets in Phase 1 and 2. The results are not returned yet for the secondary exam.**

| Phase 2 | | |
|---|---|---|
| No Heuristics | 26 | 15 |
| Distributed Repr | 28 | 13 |
| Heuristics | 28 | 13 |

**Table 7: Results for center exam in Phase 2.**

## 3. CONCLUSION

This paper describes the question answering system developed at Dublin City University for participation in the QA Lab shared task in NTCIR-11. We participated in three tasks: center exams (multiple choice) in Phase 1 and 2, and secondary exams (written) in Phase 2. Our results for center exams in Phase 1 was 77.0/100.0 which was the first among seven submissions, for center exams in Phase 2 was 72.0/100.0, and for secondary exam in Phase 2 was 72.0, and for secondary exam task in Phase 2 in terms of precision were 71.4 (UTokyo), 62.5 (KyotoU), 71.8 (Hokkaido), 62.7 (Waseda), and 80.0 (Chuo).

We built a QA system where we use our type-I semantic parser to acquire meaning representation, or case frame graphs, from history textbooks using commonsense knowledge. We used distributed representation for OOV words and missing assertions. We added prototype functionality for handling implicit arguments/relations, causality analysis, time analysis, and temporal order analysis by heuristics.

## 4. ACKNOWLEDGMENTS

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of the CNGL at Dublin City University. We would also like to thank the Irish Centre for High-End Computing.

## 5. REFERENCES

[1] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet project. *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, 1998.

[2] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *In Conference on*

*Empirical Methods in Natural Language Processing (EMNLP) 2014*, 2014.

[4] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: the second generation. *International Joint Conference on Artificial Intelligence*, 2011.

[5] C. Fellbaum. Wordnet: An electronic lexical database. *MIT Press Cambridge. MA*, 1998.

[6] R. Ge and R. J. Mooney. A statistical semantic parser that integrates syntax and semantics. *Association for Computational Linguistics*, pages 9–16, 2005.

[7] M. Gerber and J. Chai. Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics*, 38:755–798, 2012 2012.

[8] A. Haghighi, A. Ng, and C. D. Manning. Robust textual inference via graph matching. *In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2005)*, 2005.

[9] D. Kawahara and S. Kurohashi. Japanese case frame construction by coupling the verb and its closest case component. *In Proceedings of the Human Language Technology Conference (HLT2001)*, pages 204–210, 2001.

[10] D. Kawahara and S. Kurohashi. Acquiring reliable predicate-argument structures from raw corpora for case frame compilation. *In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC2010)*, pages 1389–1393, 2010.

[11] S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. Improvements of japanese morphological analyzer juman. *In Proceedings of The International Workshop on Sharable Natural Language Resources*, 1994.

[12] P. Liang. Lambda dependency-based compositional semantics. *CoRR*, abs/1309.4408, 2013.

[13] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. *CoRR*, abs/1109.6841, 2011.

[14] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *ArXiv:1309.4168*, 2013.

[15] T. Okita, A. H. Vahid, A. Way, and Q. Liu. The dcu terminology translation system for the medical query subtask at wmt14. *In Proceedings of the Workshop on Statistical Machine Translation*, 2014.

[16] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems (NIPS)*, pages 1410–1418, 2009.

[17] S. Riedel, L. Yao, B. M. Marlin, and A. McCallum. Relation extraction with matrix factorization and universal schemas. *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, 2013.

[18] K. Sagae and J. Tsujii. Shift-reduce dependency dag parsing. *In Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 753–760, 2008.

[19] Y. Shen, X. he, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. *WWW 2014*, April 2014.

[20] H. Shibuki, K. Sakamoto, Y. Kano, T. Mitamura, M. Ishioroshi, K. Itakura, D. Wang, T. Mori, and N. Kando. Overview of the ntcir-11 qa-lab task. *NTCIR-11 Proceedings*, 2014.

[21] R. Socher, J. Pennington, E. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, 2011.

[22] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, 2013.