BnO at NTCIR-10 RITE: A Strong Shallow Approach and an Inference-based Textual Entailment Recognition System

Ran Tian
National Institute of Informatics, Japan
tianran@nii.ac.jp

Yusuke Miyao
National Institute of Informatics, Japan
yusuke@nii.ac.jp

Takuya Matsuzaki
National Institute of Informatics, Japan
takuya-matsuzaki@nii.ac.jp

Hiroyoshi Komatsu
hiroyoshi.komat@gmail.com

ABSTRACT

The BnO team participated in the Recognizing Inference in TExt (RITE) subtask of the NTCIR-10 Workshop [5]. This paper describes our textual entailment recognition system with experimental results for the five Japanes subtasks: BC, MC, EXAMBC, EXAM-SEARCH, and UnitTest. Our appoach includes a shallow method based on word overlap features and named entity recognition; and a novel inference-based approach utilizing an inference engine to explore relations among *algebraic forms of sets*, which are computed from a tree representation similar to the depency-based compositional semantics.

Team Name

Beagle and Onion (BnO)

Subtasks

BC, MC, EXAMBC, EXAM-SEARCH, and UnitTest (Japanese)

Keywords

NTCIR, RITE, textual entailment, dependency-based compositional semantics, inference engine

1. INTRODUCTION

The recognition of textual entailment (RTE) is known as a hard NLP task, because there are so many complexities and possiblilities to incorporate syntacitc/semantic representations and a varitety of linguistic knowledge to make a deep semantic analyzing system, yet the result does not always outperform a simple approach like word overlap measure. In this paper, we describe the approach adopted by our RTE system, which is a two-stage classifier utilizing both shallow features and deep semantic analysis. A strong shallow system, which is a linear classifier based on a two-dimensional word overlap feature and a named-entity feature, is first constructed; the output of this shallow classifier is then brought to the second stage, being regarded as a feature to another linear classifier, which incorporates the first-stage-output with some deep semantic features, extracted from the deduction process of an inference engine.

We organize this paper as follows. In §2 we describe our shallow system. In §3 we overview the theory leading us to a deep semantic analyzing framework. In §4 we describe our inference-based system, and in §5 we discuss the experiment results. We conclude this paper in §6.

2. SHALLOW APPROACH

For a pair of text **T** and hypothesis **H**, the shallow system does the following process:

1. Chunking: we use Cabocha [3] to divide **T** and **H** into chunks.
2. Function words: for each chunk, we use Tsutusji Function Words Dictionary [6] and do max-length matching from the right, to trim the function words off.
3. Content words: from the result obtained in Step 2, we use NihongoGoiTaikei[1] and Wikipedia page titles to do max-length matching from the left, dividing the content part of each chunk into several content words.
4. For each content word c in **H**, judge if c is a named entity. In our system, time and number expressions recognized by normalizeNumexp[2], entries with POS tag [固] in NihongoGoiTaikei, or any entry which cannot be found in NihongoGoiTaikei but can be found in Wikipedia, are all regarded as named entities.
5. For each content word c in **H**, find if there is synonym of c in **T**. The knowledge of synonyms is extracted from NihongoGoiTaikei, Wikipedia redirect, BunruiGoiHyo[3] and Japanese WordNet [1].

Once we made the above process, features are defined by:

- $f_1 = 1$ if every named entity in **H** has a synonym in **T**. Otherwise $f_1 = 0.1$.
- $f_2 = f_1 \cdot \log(L_H + 1)$, where L_H is the length of the content words list l_H of **H**.
- $f_3 = f_1 \cdot \log(D_H + 1)$, where D_H is the number of words in l_H that have found their synonyms in **T**.

[1] http://www.kecl.ntt.co.jp/icl/lirg/resources/GoiTaikei/

[2] http://www.cl.ecei.tohoku.ac.jp/~katsuma/software/normalizeNumexp/

[3] http://www.ninjal.ac.jp/archives/goihyo/

BC	EXAMBC
$\gamma = 0.83$	$\gamma = 1.48$
$\alpha = 0.47$	$\alpha = 0.71$

Table 1: Learned parameters γ and α.

Two ideas appeared in the feature design are worth mentioning:

1. When a named entity is missing, we can be pretty sure that this is a negative example; however such examples are quite rare, a usual 0-1 feature of named entity missing may not get an appropriate weight by learning. Thus we ensure this prior knowledge by multiply f_1 to f_2 and f_3: if named entity is missing, the situation is almost the same, no matter how much word overlap is.

2. The pair (f_2, f_3) represents a two-dimensional word overlap feature, the log functions in f_2 and f_3 reflect the following assumption: we assume the textual entailment relation being classified by the formula

$$\frac{(D_H + 1)^\gamma}{(L_H + 1)} > \alpha.$$

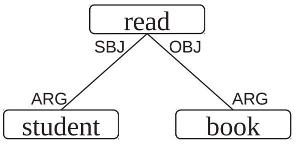
   The parameter γ and α are then learned by the linear classifier, through features f_2 and f_3. Precisely, if the learned splitting plane can be normalized to $w_1 f_1 - f_2 + w_3 f_3 > b$, then it is equivalent to the above formula by $\gamma = w_3$ and $\alpha = \exp(b - w_1 f_1)$.

The intuition behind the introduction of an additional factor γ, other than the usual word overlap ratio, is that: the making of the dataset may have some unintended bias affected by the length of hypothesis vs. the word overlap ratio. For example, given a text **T**, it may be more difficult to make a positive pair with a long **H**; or we can say there are more options to make a negative pair if **H** is longer, given the same word overlap ratio. Conversely, if we start from **H**, like the data making process of EXAMBC: **H** is taken from entrance exam questions, while corresponding **T** is extracted from textbooks or wikipedia, then it may be more likely to be a positive example if **H** is longer but still has a high rate of word overlap. Actual experiments support this intuition, as the learned parameter γ shown in Table 1 satisfy $\gamma < 1$ for BC-dataset and $\gamma > 1$ for EXAMBC-dataset. And these three features also perform very well on test data.

3. DCS-TREE REPRESENTION

Dependency-based compositional semantics (DCS) [4] was originally developed as a natural language interface for database queries. Semantics of natural language expressions are represented by DCS trees, which are rooted trees with labeled nodes and edges, thus resembling dependency trees. Nodes are labeled with predicates, each of which corresponds to a table in a relational database, and edges are labeled with input and output roles, which denote columns of tables, or intuistively, arguments of predicates. Extra markers may be added to deal with linguistic phenomena such as quantifiers and superlatives (Figure 1). An algorithm was presented for converting this representation into direct answers using a given relational database.

The requirement of a relational database in the original DCS limited the application scope of this framework; we have developed a new framework which eliminates this restriction: we do not require an explicit database, instead we translate a tree representation of natural language expression into some *algebraic forms*, and we explore the relations among these forms by an inference engine.



$$\mathfrak{I}_{\text{read}} \cap \mathfrak{I}_{\text{student}} \times \mathfrak{I}_{\text{book}}$$

Figure 1: The DCS tree for "students read books", and its corresponding algebraic form.

For example, the DCS tree of the sentence "students read books" are shown in Figure 1, with its corresponding algebraic form shown below. If we assume this sentence is true, i.e. $\mathfrak{I}_{\text{read}} \cap \mathfrak{I}_{\text{student}} \times \mathfrak{I}_{\text{book}} \neq \emptyset$, then we can imply "there is a student", or $\mathfrak{I}_{\text{student}} \neq \emptyset$. This inference precedure is done by applying the following axioms to the former algebraic form and deriving the latter: (i) $A \cap B \subset A$, (ii) $A \subset B$ & $A \neq \emptyset \Rightarrow B \neq \emptyset$, and (iii) $A \times B \neq \emptyset \Rightarrow A \neq \emptyset$.

The motivations for us to use DCS trees as semantic representations are:

1. DCS trees are precise semantic representations, yet resemble to dependency trees. Thus we can easily obtain DCS trees from semantically annotated dependency parses, and use them to do logical inferences applying structural knowledge.

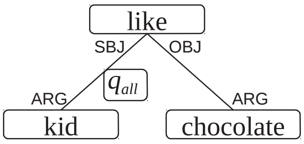
2. DCS trees represent a limited range of first order logical expressions, which is characterized by the logical system of algebraic forms. Logical inference on algebraic forms can be done fast, compared to the traditional first oreder predicate logic.

3. Algebraic forms inherit most of the tree structures appeared in natural language, which enables us to dynamically generate missing knowledge via linguistic intuitions, efficiently making the inference process go further even when the lack of prior knowledge interferes with inference chains.

Formally, assume the following sets are given:

- $\mathcal{P}$, the set of predicate labels.
- $\mathcal{R}$, the set of semantic roles.
- $\mathcal{S}$, the set of selection markers.
- $\mathcal{Q}$, the set of quantification markers.

A DCS tree $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ in our framework is defined as a rooted tree, where each node $\sigma \in \mathcal{N}$ is labeled with a predicate $p \in \mathcal{P}$ and each edge $(\sigma, \sigma') \in \mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is labeled with a pair of semantic roles $(r, r') \in \mathcal{R} \times \mathcal{R}$. Furthermore, for each node σ we can optionally assign a selection marker $\mathfrak{s} \in \mathcal{S}$, and for each edge (σ, σ') we can optionally assign a quantification marker $q \in \mathcal{Q}$.



$$\mathfrak{q}^{\mathsf{SBJ}}_{all}(\pi_{\mathsf{SBJ}}(\mathfrak{I}_{\text{like}} \cap \mathfrak{W} \times \mathfrak{I}_{\text{chocolate}}), \mathfrak{I}_{\text{kid}})$$

Figure 2: The DCS tree for "all kids like chocolates", and its corresponding algebraic form.

And algebraic forms generated from a DCS tree are symbol $\mathfrak{W}$ and symbols $\mathfrak{I}_p$ corresponding to predicate p, joined by set operators $\cap$, $\times$, π, the selection operator $\mathfrak{s} \in \mathcal{S}$ and the division operator $\mathfrak{q}^r$ corresponding to quantifier $q \in \mathcal{Q}$ and semantic role $r \in \mathcal{R}$.

Examples are shown in Figure 1 and Figure 2. We discuss our RTE system utilizing this semantics analyzing framework in the next section.

4. INFERENCE-BASED APPROACH

In our inference-based system, we first apply Cabocha [3] and Syncha [2] to obtain predicate argument structures. Then we use some simple rules to augment semantic roles; the semantic roles we identify are: SBJ, OBJ, IOBJ, ARG, THG, TIME, LOC, R1 and R2. THG is used to represent events such as the "fact" that "students read books," and R1 and R2 are used to represent any kind of directed binary relations, including possession, purpose, etc.

Linguistic knowledge we used are synonym, hypernym, and antonym relations extracted from: NihongoGoiTaikei, Wikipedia, Japanese WordNet, and Kojien dictionary.[4] While these knowledge has enabled us to establish correspondences between single words, in order to compensate for other missing knowledge such as paraphrases, we use the following method to inject on-the-fly knowledge into the inference process:

- For a pair of text **T** and hypothesis **H**, let $\mathcal{T}_T$ and $\mathcal{T}_H$ be the DCS trees of **T** and **H**, respectively. do:

1. According to statements proved by the inference engine, for each node in $\mathcal{T}_H$ find its semantic correspondences in $\mathcal{T}_T$.
2. With the correspondencies drawn between nodes in $\mathcal{T}_H$ and $\mathcal{T}_T$, we apply syntactically motivated heuristics to generate new entailment rules on words and phrases with some confidence value.
3. Add the most confident new rule to the knowledge base, and try to prove statements of $\mathcal{T}_H$, using $\mathcal{T}_T$ and the updated knowledge base.
4. Evaluate Step 3 using a cost function which is dependent on the confidence of the new rule and the number of newly proved partial statements.

By repeating the above process, we obtain newly added rules and their costs. Finally, we apply a classifier that evaluates the accmulated cost of the rules to determine the entailment relation.

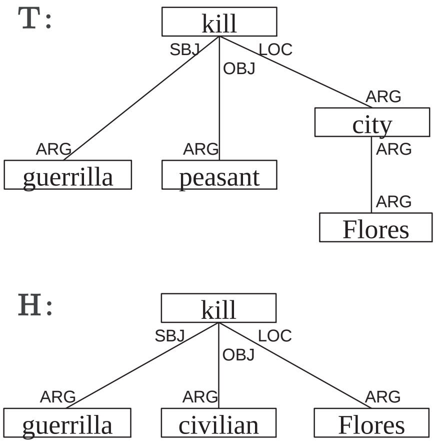


Figure 3: The DCS trees for "Guerrillas killed a peasant in the city of Flores", and "Guerrillas killed a civilian in Flores".

In the above process, we intensively use the inference engine in Step 1 and Step 4. In Step 1, by utilizing the inference engine, the process can take into account linguistic knowledge such as hypernyms, and also context information for nodes in $\mathcal{T}_T$. For example, consider the following pair: (Figure 3)

T: Guerrillas killed a peasant in the city of Flores.
H: Guerrillas killed a civilian in Flores.

The word "civilian" in **H** can find its correspondence "peasant" in **T**, if the inference engine has the knowledge that "civilian" is a hypernym of "peasant". Also, the system will regard both "Flores" and "city" in **T** as the correspondencies of "Flores" in **H**, because calculation of the DCS tree $\mathcal{T}_T$ will assign equivalent algebraic forms to the node "Flores" and "city".

In Step 4, the partial statements are algebraic forms generated from (subtrees of) $\mathcal{T}_H$, which are consequences if **H** is true. For example, we can use statements like "there is a civilian" ($\mathfrak{I}_{\text{civilian}} \neq \emptyset$) or "killed a civilian" ($\mathfrak{I}_{\text{kill}} \cap \mathfrak{W} \times \mathfrak{I}_{\text{civilian}} \times \mathfrak{W} \neq \emptyset$) as partial statements of "Guerrillas killed a civilian in Flores".

Step 2 of generating missing knowledge via linguistic intuitions and estimating their confidence values, is quite ad hoc. Roughly speaking, we applied 4 kinds of heuristics to generate missing knowledge, which are heuristics considering:

1. Two continuous nouns appeared in **H**, of which one can find its correspondence in **T** but the other one cannot. In this case there is a possibility that these two nouns are appositive, as the words "小説 (novel)" and "雪国 (Yukiguni)" in the sentence "川端康成は小説「雪国」を書いた (Kawabata Yasunari wrote the novel Yukiguni)".

2. The Japanese word "の" representing various kinds of relations. If a pair of nouns appears in both **T** and **H**, and one of them is joined by "の", we copy the

[4] http://www.iwanami.co.jp/kojien/

	shallow			inference		
	gold\sys	N	Y	gold\sys	N	Y
BC	N	285	69	N	296	58
	Y	57	199	Y	70	186
	gold\sys	N	Y	gold\sys	N	Y
EXAMBC	N	206	69	N	227	48
	Y	71	102	Y	85	88

Table 2: Confusion Matrices for BC and EXAMBC

relation in the other sentence and regard this “の” as that relation.

3. Words in **H** that cannot find their correspondencies in **T**. We may guess a correspondence regarding the words around.

4. Phrases (or DCS tree fragments) in **H** that cannot be proved by **T**. We may connect them to some parts of **T** according to word correspondencies.

Some examples are shown in the next section. The confidence for each newly generated knowledge is calculated based on word similarities using BunruiGoiHyo. We tuned this confidence function using RITE EXAM and RITE2 EXAMBC development sets, and no machine learning methods were applied. This step will be improved in the future.

5. EXPERIMENTS

We directly applied the shallow system and inference-based system to subtasks BC and EXAMBC. In our submitted runs, “run3” was the output of the shallow system (we will refer to it as system “shallow” hereinafter), while “run1” and “run2” were inference-based systems with slightly different features. We shall mainly discuss “run2” because it outperforms “run1” (“run2” is refered to as system “inference” from now on).

The confusion matrices of system “shallow” and system “inference”, tested on BC-test and EXAMBC-test data, are shown in Table 2. Not surprisingly, the “shallow” system tends to recognize more negative pairs incorrectly as “Y”, compared to the “inference” system who tends to make more mistakes on positive pairs. The classification of system “inference” mainly depends on three factors, besides the one feature which comes from the output of “shallow” system: (i) the portion of proved partial statements before any on-the-fly knowledge is injected, (ii) the portion of newly proved partial statements after some heuristic knowledge between single words is assumed, and (iii) the portion of newly proved partial statements after some on-the-fly paraphrase knowledge is generated. We show some examples in the BC-test data:

- There are 14 pairs where the shallow system outputs a “Y”, the inference system outputs an “N”, and the gold label is “Y”. One example is

  > **T**: トスカーナ大公の称号がメディチ家に授与され、フィレンツェはトスカーナ大公国の首都となった。
  > (The title Grand Duke of Tuscany is given to Medici, and Florence became the capital city of Grand Duchy of Tuscany.)
  > **H**: メディチ家は後にトスカーナ大公国の君主となった一族だ。

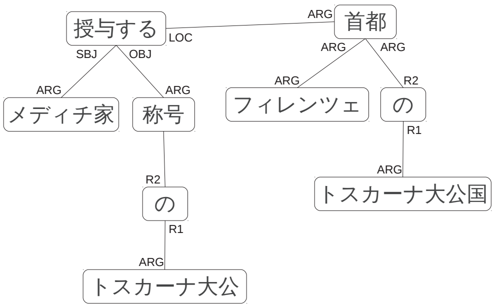


Figure 4: The DCS tree output of “トスカーナ大公の称号がメディチ家に授与され、フィレンツェはトスカーナ大公国の首都となった”.

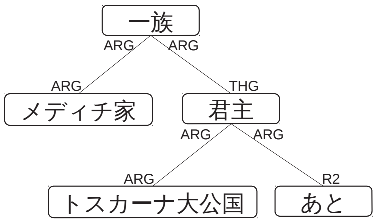


Figure 5: The DCS tree output of “メディチ家は後にトスカーナ大公国の君主となった一族だ”.

  > (The House of Medici was the family who became the monarch of Grand Duchy of Tuscany later.)

  The DCS tree output by our system for **T** and **H** are shown in Figure 4 and Figure 5 respectively. The system has the knowledge that “君主 (monarch)” is a hypernym of “トスカーナ大公 (Grand Duke of Tuscany)”。 It then guessed that “トスカーナ大公” implies “トスカーナ大公国の君主 (the monarch of Grand Duchy of Tuscany)”, because “トスカーナ大公国” and “君主” are two continuous nouns and may be appositive. Though the reason is suspicious this is a correct guess. Next the system guessed that “トスカーナ大公の称号 (The title Grand Duke of Tuscany)” is the same as “トスカーナ大公”, due to the heuristic rule on word “の”. These efforts made some progress in the inference procedure, but the systems failed to find any clue that may prove “君主となった一族 (the family who became the monarch)”。

- There are 12 pairs where shallow system outputs “Y”, inference system outputs “N”, and gold label is “N”. One example:

  > **T**: 吉本印天然素材は、1991 年 9 月、吉本興業所属の若手芸人で結成された、ダンスとお笑いをミックスしたユニットである。
  > (Yoshimoto Jirushi Tennensozai founded on Sep.1991 is a mix-unit of dance and comedy, formed by young comedians belonging to Yoshimoto Kogyo.)
  > **H**: 天然素材は，自然で作られた素材のことで

> ある。
> (Natural materials are materials that formed in nature.)

For this pair, our inference system regarded “天然素材 (Natural materials)” as a hypernym of “吉本印天然素材 (Yoshimoto Jirushi Tennensozai)”, and from this connection it guessed that “作られる (formed)” can be infered by “結成される (formed)”, which seems quite reasonable; furthermore after the assumption that “芸人で結成される (formed by comedians)” implies “自然で作られる (formed in nature)”, the system even proved the hypothesis. However the classifier finally output an “N” because the confidence of the assumption “芸人で結成される” implying “自然で作られる” is quite low while the inference progress made by this assumption is quite large - this results in a low score for the proof found by the inference system. Moreover, the portion of partial statements initially proved before any on-the-fly knowledge is very low (or 0, since only one word “天然素材” has initially found correspondences in **T**). These two factors caused the negative judgement.

- There is one pair where shallow output is “N”, inference output is “Y”, and gold label is “Y”:

> **T**: 魚肉及び鯨肉の原材料に占める重量の割合が 15%以上になると、「ソーセージ」の規格を外れ、魚肉及び鯨肉が 15%以上 50%未満なら「混合ソーセージ」、50%以上なら「魚肉ソーセージ」の規格に分類される。
> (If the portion of weight of fish and whale meat in raw materials exceeds 15%, it falls out the standard of “sausage”, and becomes “mixed sausage” when fish and whale meat is between 15% and 50%, “fish sausage” when the portion is over 50%.)
> **H**: 規格では、魚肉及び鯨肉の原材料に占める重量の割合が 50 %以上のものを「魚肉ソーセージ」としており、15 %未満の「ソーセージ」や 15 %以上 50 %未満の「混合ソーセージ」とは区別されている。
> (By standard, products with an over 50% portion of weight of fish and whale meat in raw materials falls into the category “fish sausage”, which is distinguished from “sausage” with a portion less than 15%, or “mixed sausage” of which the portion is between 15% and 50%.)

Shallow output is “N” because **H** is a long sentence. Nevertheless the inference output is “Y”, which shows the robustness of our inference system.

- There is also one pair which has a shallow output “N”, an inference output “Y”, and a gold label “N”:

> **T**: 襟カラーは，日本の詰襟型男子学生服などで用いられる。
> (Stand-up collars are used in Japanese stand-up-collar-styled male school uniforms.)
> **H**: 詰襟学生服が学校の制服として男子学生に着用されている。
> (Stand-up-collar school uniforms are worn by boy students as uniforms in school.

The inference output is “Y”, because synonyms extracted from wikipedia redirect include the pair “詰襟学生服 (Stand-up-collar school uniform) = 学生服 (school uniform)”, and the system also has the knowledge that “制服 (uniform)” is a hypernym of “学生服 (school uniform)”.

- There are 56 pairs where both shallow and inference systems output “N”, but gold label is “Y”. Some hard examples are:

> **T**: マニアとされる人々は、大衆一般への普及を目指すゼネラルオーディオなど、廉価な製品には見向きもしない。
> (People that are geeks take no notice of low-cost products like the general audios aimed at popularization to common people.)
> **H**: オーディオマニアは，ゼネラルオーディオを「廉価版製品」とみなす。
> (Audiophiles regard general audios as low-cost products.)

In this pair, the system has the knowledge that “廉価版 = 廉価 (low-cost)” and “オーディオマニア (Audiophiles) ⊂ マニア (geeks)”, but it didn’t connect “オーディオマニア” to “マニア”, because the hyponym “オーディオマニア” is in **H**.

> **T**: 日本では明治初期が木版印刷から活版印刷への移行期である。
> (The early Meiji era is the transition period from block printing to typography in Japan.)
> **H**: 日本で活版印刷が広く行われるようになるのは明治時代以降である。
> (In Japan, typography became widespreading since the Meiji era.)

This pair requires deep knowledge on the meaning of the word “移行期 (transition period)”, which we don’t have.

> **T**: 非可逆圧縮は、人間の感覚に伝わりにくい部分は情報を大幅に減らし、伝わりやすい部分の情報を多く残すように圧縮を行う。
> (Lossy compression performs the compression by greatly reducing the information from parts that hardly transmitted to human senses, while leaving much information about easily transmitted parts.)
> **H**: 多くの非可逆圧縮では人間があまり強く認識しない成分を削除することでデータを圧縮する方法がとられている。
> (A lot of lossy compressions adopt the method that removes components which are usually not strongly recognized by humans.)

In this pair, the inference system somehow guessed that “減らす (reduce)” implies “削除する (remove)”, however it failed to recognize the correspondence between “ 人間の感覚に伝わりにくい部分 (parts that hardly transmitted to human senses)” and “人間があまり強く認識しない成分 (components which are usually not strongly recognized by humans)”.

- There are 57 pairs where both shallow and inference systems output “Y”, but gold label is “N”. Some of these pairs seem controversial, for example

gold\sys	B	F	C	I
B	46	12	0	12
F	14	163	2	26
C	11	21	2	27
I	13	46	4	149

Table 3: Confusion Matrices for MC

> **T**: ルートヴィヒ・ウィトゲンシュタインはその著書『哲学探究』のなかで、「ゲーム」という語をとりあげ、「ゲーム」と呼ばれている全ての外延（対象）を特徴づけるような共通の内包（意義）は存在せず、実際には「勝敗が定まること」や「娯楽性」など部分的に共通する特徴によって全体が緩くつながっているに過ぎないことを指摘し、これを家族的類似と名付けた。
> (In his book “Philosophical Inquiry”, Ludwig Wittgenstein picked up the word “game”...)
> **H**: 『哲学探究』はルートヴィヒ・ウィトゲンシュタインによって執筆された。
> (“Philosophical Inquiry” was written by Ludwig Wittgenstein.)

and

> **T**: 荻窪郵便局は、東京都杉並区にある郵便局である。
> (Ogikubo post office is a post office located in Suginami-ku, Tokyo.)
> **H**: 荻窪とは、東京都杉並区にある地名である。
> (Ogikubo is the name of a place in Suginami-ku, Tokyo.)

Others somehow need the knowledge of “words that shouldn’t be mistaken”, not limited to named entities. For example in the pair

> **T**: 藤間一男は、ビリヤード場を経営する家庭の長男として生まれた。
> (Fujima Kazuo was born as the eldest son of a family who runs a billiard hall.)
> **H**: 藤間一男は、ビリヤード場を経営していた。
> (Fujima Kazuo runs a billiard hall.)

we need the knowledge that “家庭 (family)” and “家庭の長男 (the eldest son of a family)” are not the same. And in

> **T**: 操縦手は、スペースシャトルの操縦を担当する宇宙飛行士のことであり、パイロット宇宙飛行士の資格を得て飛行する。
> (A pilot is an astronaut who takes the control of a space shuttle, and who flys by given a pilot astronaut qualification.)
> **H**: 船長は、スペースシャトルの操縦を担当する宇宙飛行士だ。
> (Captain is the astronaut who take control of a space shuttle.)

we need to know that “船長 (captain)” is disjoint to “操縦手 (pilot)”.

textbook			Wikipedia		
gold\sys	N	Y	gold\sys	N	Y
N	174	101	N	170	105
Y	101	72	Y	90	83

Table 4: Confusion Matrices for EXAM-SEARCH

	Search Prec.	Search Rec.
textbook	22.32	10.99
Wikipedia	26.56	13.08

Table 5: Search evaluation for EXAM-SEARCH

As for subtasks MC, EXAM-SEARCH and UnitTest, we only used inference-based features. For MC, we extract features from both the inference processes of not only assuming **T** and trying to prove **H**, but also assuming **H** and trying to prove **T**. Since the examples with label “C” and label “B” are rare in the training data, we trained the classifier assigning these examples 4 times of weight. Also, since there is no intuitive reason for using a linear kernel for this multi-class task, we also tried the rbf kernel. The rbf kernel (run3) achieved better F1-scores on B, F and I classes compared to linear kernel (run2), so we show the confusion matrix of run3 (rbf kernel) in Table 3. The system is not good at detecting the “C” label, maybe partly because of the lack of training data, and partly because the criteria for a “C” label is not very clear – usually by directly performing logical inferences on these pairs a contradiction cannot be obtained, but additional world knowledge is neccessary.

For EXAM-SEARCH, corresponding to each **H** there are 5 passages extracted from both the textbook and Wikipedia, respectively, provided by organizers. We trained a model using EXAMBC-dev data, and chose one passage of the highest score as **T**. The confusion matrix of system outputs, and the precision and recall for the chosen passage, are shown in Table 4 and Table 5. It seems that Wikipedia has a better search result.

For UnitTest, we used BC-dev data for training. Precisions and recalls for each category are shown in Table 6. Most mistakes occur on negative examples categorized as “synonymy:phrase” and “entailment:phrase”. Our system with only inference-based features didn’t get the top result, but still outperform the baseline, which shows our inference-based system actually helps to recognize textual entailments.

6. CONCLUSIONS

We described our approach to recognition of textual entailment in the NTCIR-10 RITE-2 shared task, which includes a strong shallow system and an inference-based deep semantic analyzing framework. Experiments show a hybrid of inference-based with the shallow system can achieve high accuracies, and we have discussed some examples of when the inference-based features does or does not work.

Besides the examples listed above, there also found many errors introduced by processes in the inference-based system, including erroneous DCS trees, false knowledge, weird behaviour of on-the-fly heuristics, inaccurate confidence function, etc. Refinement of this framework, collection and integration of more knowledge, should be the immediate future works.

7. REFERENCES

[1] F. Bond, T. Baldwin, R. Fothergill, and K. Uchimoto. Japanese semcor: A sense-tagged corpus of japanese. In

Category	Y:Prec	Y:Rec	N:Prec	N:Rec
case alternation	7/ 7	7/ 7	0/ 0	0/ 0
inference	2/ 2	2/ 2	0/ 0	0/ 0
spatial	0/ 0	0/ 1	0/ 1	0/ 0
implicit relation	18/ 18	18/ 18	0/ 0	0/ 0
list	3/ 3	3/ 3	0/ 0	0/ 0
disagree:lex	0/ 1	0/ 0	1/ 1	1/ 2
synonymy:phrase	27/ 27	27/ 35	0/ 8	0/ 0
meronymy:lex	1/ 1	1/ 1	0/ 0	0/ 0
apposition	0/ 0	0/ 1	0/ 1	0/ 0
modifier	41/ 41	41/ 42	0/ 1	0/ 0
transparent head	1/ 1	1/ 1	0/ 0	0/ 0
synonymy:lex	9/ 9	9/ 10	0/ 1	0/ 0
nominalization	1/ 1	1/ 1	0/ 0	0/ 0
coreference	4/ 4	4/ 4	0/ 0	0/ 0
disagree:phrase	0/ 5	0/ 0	20/ 20	20/ 25
temporal	1/ 1	1/ 1	0/ 0	0/ 0
disagree:modality	0/ 1	0/ 0	0/ 0	0/ 1
entailment:phrase	31/ 31	31/ 45	0/ 14	0/ 0
disagree:temporal	0/ 1	0/ 0	0/ 0	0/ 1
hypernymy:lex	3/ 3	3/ 3	0/ 0	0/ 0
scrambling	15/ 15	15/ 15	0/ 0	0/ 0
clause	14/ 14	14/ 14	0/ 0	0/ 0
relative clause	8/ 8	8/ 8	0/ 0	0/ 0

Table 6: Evaluation on UnitTest

Proceedings of GWC-2012, 2012.

[2] R. Iida and M. Poesio. A cross-lingual ilp solution to zero anaphora resolution. In *Proceedings of ACL-HLT 2011*, 2011.

[3] T. Kudo and Y. Matsumoto. Japanese dependency analyisis using cascaded chunking. In *Proceedings of CoNLL 2002*, 2002.

[4] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. In *Proceedings of ACL 2011*, 2011.

[5] Y. Watanabe, Y. Miyao, J. Mizuno, T. Shibata, H. Kanayama, C.-W. Lee, C.-J. Lin, S. Shi, T. Mitamura, N. Kando, H. Shima, and K. Takeda. Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. In *Proceedings of the 10th NTCIR Conference*, 2013.

[6] 松吉俊, 佐藤理史, and 宇津呂武仁. 日本語機能表現辞書の編纂. In **自然言語処理**, 2007.