Overview of the NTCIR-12 Short Text Conversation Task

Lifeng Shang
Noah's Ark Lab of Huawei, Hong Kong
Shang.Lifeng@huawei.com

Tetsuya Sakai
Waseda University, Japan
tetsuyasakai@acm.org

Zhengdong Lu
Noah's Ark Lab of Huawei, Hong Kong
Lu.Zhengdong@huawei.com

Hang Li
Noah's Ark Lab of Huawei, Hong Kong
HangLi.HL@huawei.com

Ryuichiro Higashinaka
Nippon Telegraph and Telephone Corporation, Japan
higashinaka.ryuichiro@lab.ntt.co.jp

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

ABSTRACT

We describe an overview of the NTCIR-12 Short Text Conversation (STC) task, which is a new pilot task of NTCIR-12. STC consists of two subtasks: a Chinese subtask using post-comment pairs crawled from Weibo[1], and a Japanese subtask providing the IDs of such pairs from Twitter[2]. Thus, the main difference between the two subtasks lies in the sources and languages of the test collections. For the Chinese subtask, there were a total of 38 registrations, and 16 of them finally submitted 44 runs. For the Japanese subtask, there were 12 registrations in total, and 7 of them submitted 25 runs. We review in this paper the task definition, evaluation measures, test collections, and the evaluation results of all teams.

Keywords

artificial intelligence, dialogue systems, evaluation, information retrieval, natural language processing, social media, test collections.

1. INTRODUCTION

Achieving natural language conversation between humans and computers is one of the most challenging artificial intelligence (AI) problems. It involves language understanding, reasoning, planning, and the use of a knowledge base. Although a significant amount of research has been done, the progress in solving this problem is unfortunately still quite limited. One of the major reasons for this is the lack of a large volume of real conversation data.

In the NTCIR-12 Short Text Conversation pilot task, we consider a much simplified version of the original problem: one round of conversation formed by two short texts, with the former being an initial post from a user and the latter being a comment given by the computer. We refer to this as a short text conversation (STC). Because of the extremely large amount of short text conversation data available on social media such as Twitter and Weibo, we anticipate that significant progress can be made in the research on this problem with the use of big data, much like what has happened in machine translation, community question answering, and other fields.

One simple approach to STC is to take it as an information retrieval (IR) problem, maintain a large repository of short text conversation data, and develop a conversation system mainly based on IR technologies. The basic idea of this approach is graphically shown in Figure 1. Given a new post **A**, the system searches the repository to return the most suitable comment. The comments in the repository were originally posted in response to some posts other than **A**, but we assume that they can be reused as a reasonable comment to **A**. That is, rather than pursuing generation-based STC (i.e., generating suitable comments given an initial post from the user) [16], we tackle the simpler problem of retrieval-based STC. With advanced IR technologies and big data, even retrieval-based STC systems may eventually behave like humans in each round of conversation.

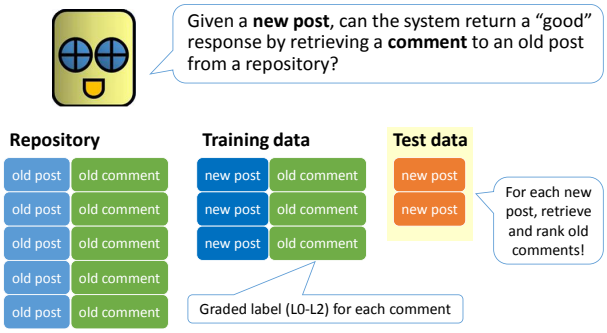

Figure 1: Approaching STC as an IR problem.

The key research question which we would like to address here is: given a new post, can an appropriate (i.e., "human-like") comment be returned by searching a post-comment repository? What are the challenges and limitations of retrieval-based STC? There are many applications that can benefit from the research on STC, for example, chatbots on websites, automatic message reply on mobile phones, and voice assistants such as Siri. The research on it will also shed light on language understanding and human behavior studies.

Although some tasks in previous TREC (Text Retrieval Conference) and NTCIR were similar to STC, there were some obvious differences between them. One related task is the TREC Microblog track [8], and the other related task is NTCIR-8 CQA (Community QA Test Collection) [6]. The purpose of the TREC Microblog Track is to find the most recent but relevant tweets to the user's query, which is similar to a traditional web search. Table 1 summarizes the difference between the Microblog Track and our proposed STC task. The goal of the NTCIR-8 CQA task is to identify the best answer or good answers for a given question from all the answers to the question within a CQA session. Table 2 summarizes the difference between the CQA task and STC task. In the STC task, each

[1]http://www.weibo.com/.
[2]https://twitter.com/.

Table 1: Difference between TREC Microblog Track and STC Task

	TREC Microblog Track	NTCIR-12 STC Task
Objective	To find the most recent but relevant tweets to the user's query	To find the most appropriate comments for a new query post
Dataset	Twitter with English	Sina Weibo with Chinese and Twitter with Japanese
Retrieval Repository	A set of tweets	A set of post-comment pairs

Table 2: Difference between NTCIR-8 CQA Task and STC Task

	NTCIR-8 CQA Task	NTCIR-12 STC Task
Objective	To identify the best answer or good answers for a question from all the answers to the question within a CQA session	To find the most appropriate comments for a new post from all the historical comments in the social media
Dataset	Japanese Yahoo! Answers	Sina Weibo with Chinese and Twitter with Japanese
Query Type	Only questions	Any type of sentences including questions
Retrieval Repository	The real answers to each question within a CQA session (Strictly speaking, it is not a retrieval task, but a classification task.)	A set of post-comment pairs

instance consists of a post-comment pair, while in the CQA task, each instance consists of a question-answer pair. The posts tend to be longer than the comments in STC, while the answers tend to be longer than the questions in CQA. The answers in CQA must be replies to the questions, which involves mostly knowledge; while the comments in STC need only be explanations, illustrations, or criticisms of the posts, which is more about appropriateness or being human-like.

There are many open questions as well as challenges with regard to STC. When performing retrieval-based STC, we need to consider matching between post and comment in terms of topical relevance. In addition, we may also need to consider matching between post and comment in terms of speech act, sentiment, entity relation, and discourse structure. Determining how to model the factors and how to enhance the accuracy based on the factors in STC are open and challenging issues.

Fifty groups registered to take part in the STC task, and we ultimately received 44 runs from 16 teams in the Chinese subtask and 25 runs from 7 teams in the Japanese subtask. The group name, organization and the number of runs submitted to the Chinese and Japanese subtasks are listed in Tables 3 and 7, respectively.

The reminder of this paper is organized as follows. In section 2, we describe the Chinese subtask from the aspects of task definition, evaluation measures, dataset collection, and evaluation results. In section 3, we describe the details of the Japanese subtask. Section 4 concludes the paper and mentions future work.

Table 3: Organization and number of submitted runs of participating groups in STC Chinese subtask

Group ID	Organization	#runs
Nders	NetDragon WebSoft Inc.	1
BUPTTeam	Beijing University of Posts and Telecommunications	5
CYUT	Chaoyang University of Technology	1
Grad1	Institute of Information Engineering, CAS	1
HITSZ	Harbin Institute of Technology Shenzhen Graduate School	3
ICL00	Peking University	1
ITNLP	Harbin Institute of Technology	3
KGO	University of Tokushima	2
MSRSC	Microsoft Research Asia	3
OKSAT	Osaka Kyoiku University	5
picl	Peking University	2
PolyU	The Hong Kong Polytechnic University	3
splab	Shanghai Jiaotong University	3
USTC	University of Science and Technology of China	5
uwnlp	University of Waterloo	5
WUST	Wuhan University of Science and Technology	1

2. CHINESE SUBTASK

2.1 Task Definition

At NTCIR12, STC is defined as an IR problem, i.e., retrieval-based STC. A repository of post-comment pairs from Sina Weibo is prepared. A typical example of post-comment pairs is shown in Figure 2. Each participating team receives the repository in advance.

- In the **training** period, all participants build their own conversation system based on the received repository and IR technologies, and using labeled post-comment pairs as training data.
- In the **test** period, each team is given some test posts that have been held out from the repository. Each team is asked to provide a ranked list of ten results (comments) for each query. The comments must be those from the repository.
- In the **evaluation** period, the results from all participating teams are pooled and labeled by humans. Graded relevance IR measures are used for evaluation.

The original Web texts are in Chinese. To help non-Chinese participants, we provide English translations of the original texts using machine translation. Non-native speakers can get a rough idea of

Post	创新工场三年庆，在我们的「智慧树」会议室。 Today is the 3-year anniversary of Innovation Works. We are in the meeting room named Tree of Wisdom.
Comment 1	时间过得真快，创新工场都3年了！周年庆快乐！ How time flies; Innovative Works is three years old! Happy Anniversary!
Comment 2	小小智慧树，快乐做游戏，耶！ Little Wisdom Tree, happy games, yeah!
Comment 3	会议室挺气派，顶一个！ The meeting room is quite impressive, the best one!

Figure 2: A typical example of Sina Weibo post and the comments it received. Here, we only show three comments of the post and thus we get three post-comment pairs. The original text is in Chinese, and we also translate it into English. The translations of this table are performed by the authors.

the content from the translations and can still participate in the task.

2.2 Evaluation Measures

The official evaluation measures of the STC task [15] are graded-relevance IR evaluation measures for *navigational* intents [1]. This is because a human-computer conversation system that can respond naturally to a natural language post would usually require exactly one good comment. Below, we define the official measures and clarify the relationships among them. We compute these evaluation measures using the `NTCIREVAL` tool[3].

2.2.1 nG@1

Let $g(r)$ denote the *gain* of a document (i.e., a comment) retrieved at rank r. Throughout this paper, we let $g(r) = 2^2 - 1 = 3$ if the document is L2-relevant, $g(r) = 2^1 - 1 = 1$ if it is L1-relevant and $g(r) = 0$ if it is not relevant (i.e., L0). For a given topic (i.e., a post), an *ideal ranked list* is constructed by listing up all L2-relevant documents followed by all L1-relevant ones. Let $g^*(r)$ denote the gain of a comment at rank r in the ideal list. Normalized Gain at Rank 1 is defined as follows:

$$nG@1 = \frac{g(1)}{g^*(1)} . \quad (1)$$

This is a crude measure, in that it only looks at the top-ranked document, and that, in our setting, it only takes three values: 0, 1/3 or 1.

2.2.2 nERR@10

Expected Reciprocal Rank (ERR) [3] is a popular measure with a *diminishing return* property; once a relevant document is found in the list, the value of the next relevant document in the same list is guaranteed to go down. Hence, the measure is suitable for navigational intent where the user does not want redundant information. ERR assumes that the user scans a ranked list from top to bottom, and that the probability that the user is satisfied with the document at rank r is given by $p(r) = \frac{g(r)}{2^H}$, where H denotes the highest relevance level for a test collection (2 in our case). Hence, in our setting, $p(r) = 3/4$ if the document at rank r is L2-relevant, $p(r) = 1/4$ if it is L1-relevant, and $p(r) = 0$ if it is not relevant. The probability that the user reaches as far as rank r and then stops scanning the list (due to satisfaction) is given by:

$$Pr_{ERR}(r) = p(r) \prod_{k=1}^{r-1} (1 - p(k)) , \quad (2)$$

and the *utility* of the ranked list to the user who stopped at r is computed as $1/r$ (i.e., only the final document is considered to be useful). Therefore, ERR is defined as:

$$ERR = \sum_r Pr_{ERR}(r) \frac{1}{r} . \quad (3)$$

ERR is known to be a member of the *Normalized Cumulative Utility* (NCU) family [14], which is defined in terms of a stopping probability distribution over ranks ($Pr_{ERR}(r)$ in this case) and the utility at a particular rank ($1/r$ in this case).

As ERR is not normalized, it may be normalized using the aforementioned ideal list. Let $p^*(r)$ denote the stopping probability at rank r in an ideal list, and let $Pr^*_{ERR}(r)$ be defined in a way similar to Eq 2. Normalized ERR at a cutoff l is given by:

$$nERR@l = \frac{\sum_{r=1}^{l} Pr_{ERR}(r)(1/r)}{\sum_{r=1}^{l} Pr^*_{ERR}(r)(1/r)} . \quad (4)$$

The primary measure of STC is nERR@10. Note that, when $l = 1$ in Eq. 4,

$$nERR@1 = \frac{Pr_{ERR}(1)}{Pr^*_{ERR}(1)} = \frac{p(1)}{p^*(1)} = \frac{g(1)/2^H}{g^*(1)/2^H} = \frac{g(1)}{g^*(1)} = nG@1 . \quad (5)$$

That is, nG@1 can alternatively be referred to as nERR@1.

2.2.3 P^+

P^+, proposed at AIRS 2006 [10], is another evaluation measure designed for navigational intent. Like ERR, it is a member of the NCU family. Given a ranked list, let r_p be the rank of the document that has the highest relevance level in that particular list (which may or may not be H, the highest relevance level for the entire test collection) *and* is closest to the top of the list. For example, if the ranked list has L2-relevant documents at ranks 2 and 5, and an L1-relevant document at rank 1, then $r_p = 2$; if the ranked list does not contain any L2-relevant documents but has L1-relevant documents at ranks 3 and 5, then $r_p = 3$. The basic assumption behind P^+ is that no user will ever go beyond r_p: the *preferred rank*.

P^+ assumes that the distribution of users who will stop scanning the ranked list at a particular rank is uniform over all relevant documents at or above r_p. For example, if there is an L1-relevant document at rank 1 and an L2-relevant document at rank $r_p = 2$, then it is assumed that 50% of users will stop at rank 1, and the other 50% will stop at rank 2. More generally, let $I(r) = 0$ if the document at rank r is not relevant and $I(r) = 1$ otherwise; the stopping probability at each relevant document at or above r_p is assumed to be[4] $1/\sum_{r=1}^{r_p} I(r)$.

While ERR uses the *reciprocal rank* $(1/r)$ to measure the utility of a ranked list for users who stopped at rank r, P^+ employs the *blended ratio* $BR(r)$ just like *Q-measure* [14]:

$$BR(r) = \frac{\sum_{k=1}^{r} I(k) + \sum_{k=1}^{r} g(k)}{r + \sum_{k=1}^{r} g^*(k)} . \quad (6)$$

Note that *precision* based on binary relevance is given by $P(r) = \sum_{k=1}^{r} I(k)/r$, while *normalized cumulative gain* [7] based on graded

[3] `http://research.nii.ac.jp/ntcir/tools/ntcireval-en.html`

[4] Note that *Average Precision* and *Q-measure* assume a uniform distribution over *all* relevant documents, so that the stopping probability of each relevant document is $1/R$, where R is the total number of relevant documents [14].

Table 4: Statistics of dataset for Chinese subtask

Retrieval Repository	#posts	196,495
	#comments	4,637,926
	#original pairs	5,648,128
Labeled Data	#posts	225
	#comments	6,017
	#labeled pairs	6,017
Test Data	#query posts	100

relevance is given by $nCG(r) = \sum_{k=1}^{r} g(k) / \sum_{k=1}^{r} g^*(k)$. $BR(r)$ combines these two measures; the r in the denominator of Eq. 6 discounts documents based on ranks.

Finally, P$^+$ is defined as follows. If the ranked list does not contain any relevant documents, let $P^+ = 0$. Otherwise,

$$P^+ = \sum_r Pr_+(r) BR(r) = \frac{1}{\sum_{r=1}^{r_p} I(r)} \sum_{r=1}^{r_p} I(r) BR(r) \,. \quad (7)$$

Here, $Pr_+(r)$ denotes the aforementioned uniform stopping probability distribution over relevant documents ranked at or above rank r_p.

Consider a ranked list that contains one document only. If this document is not relevant, $P^+ = 0$ by definition. If it is relevant, then $r_p = 1$ and $I(1) = 1$, and therefore

$$P^+ = \frac{1}{I(1)} I(1) BR(1) = BR(1) = \frac{I(1) + g(1)}{1 + g^*(1)} = \frac{1 + g(1)}{1 + g^*(1)} \,, \quad (8)$$

which is very similar to the definition of nCG@1 (a.k.a. nERR@1). Also note that, regardless of the ranked list size, $P^+ = 1$ iff $r_p = 1$ *and* the top-ranked document is one of the most relevant ones for that topic.

2.3 Chinese Test Collection

2.3.1 The Weibo Corpus

Just like Twitter, Weibo has a limit of 140 Chinese characters for both posts and comments, making the post-comment pair an ideal surrogate for short-text conversation. To construct this million scale dataset, we first crawl hundreds of millions of post-comment pairs, and then clean the raw data in a similar way as suggested in a published study [17], including 1) removing trivial comments such as "wow", 2) filtering out potential advertisements, and 3) removing the comments after the first 30 ones for topic consistency.

Table 4 lists the statistics of the retrieval repository, labeled data, and query posts that we provided in the task. We collected 196,495 Weibo posts and the 4,637,926 corresponding comments, and finally obtained 5,648,128 post-comment pairs. Each post has 28 different comments on average, and one comment can be used to respond to multiple posts.

2.3.2 Training Data

In addition, we manually labeled 225 query posts, each of which has about 30 candidate comments. Note that for each selected (query) post, the labeled comments were originally posted in response to posts other than the query post. Finally, we labeled 6,017 comments as "suitable", "neutral", and "unsuitable". Here, "suitable" means that the comment is clearly a suitable comment to the post, "neutral" means that the comment can be a comment to the post in a specific scenario, while "unsuitable" means it is neither of the two former cases. The details of the labeling criteria are given in the following section 2.3.4.

2.3.3 Test Data

Using Sakai's *topic set size design* tool [13][5], we decided to create $n = 100$ test topics. According to our pilot study, this meant:

- If P$^+$ or nERR@10 is used for evaluation, this test set would achieve a minimum detectable difference of 0.10 for comparing $m = 2$ systems[6];
- If P$^+$ or nERR@10 is used for evaluation, this test set would achieve a minimum detectable range of 0.15 for comparing $m = 10$ systems; also, this test set would be expected to make the confidence interval width of the difference between any systems be 0.15 or smaller [12, 13];
- If P$^+$ or nERR@10 is used for evaluation, this test set would achieve a minimum detectable range of 0.20 for comparing $m = 50$ systems;
- If nG@1 is used for evaluation, this test set would achieve a minimum detectable range of 0.20 for comparing $m = 5$ systems.

Details can be found elsewhere [15].

We carefully select the test query posts to make the task adequate, balanced, and sufficiently challenging. Moreover, the selected query posts are not included in the retrieval repository in order to ensure that the retrieved comments are originally posted in response to posts other than the query posts. The participants are permitted to submit up to four runs for the task. In each run, a ranking list of ten comments for each test query is requested. The participants are also encouraged to rank their submitted runs by preference.

- For comparison purposes, at least one compulsory run that does not use any external evidence is requested. External evidence means evidence beyond the given dataset. For instance, this includes other data or information from Weibo, as well as other corpora, e.g. HowNet or the web.
- Beyond this, the participants are at liberty to submit manual, external runs, which could be useful to improve the quality of the test collections.

2.3.4 Relevance Assessments

We employ conventional IR evaluation methodology. All the results from participants are pooled to perform manual annotation with the `NTCIREVAL` tool. Post-comment pairs will be judged for their appropriateness as natural comments to posts. The labelers are instructed to imagine that they were the authors of the original posts and to judge whether a retrieved comment is appropriate (or useful) for an input post. Three levels are assigned to a comment, with scores from 0 to 2 (i.e., L0, L1, and L2).

To make the annotation task operable, the appropriateness of retrieved comments is judged from the following four criteria:

(1) **Coherent**: logically connected to the new post;
(2) **Topically relevant**: the topic matches that of the new post;
(3) **Context-independent**: "good or not" does not depend on situations;
(4) **Non-repetitive**: does not just repeat what the new post says;

[5] `http://www.f.waseda.jp/tetsuya/CIKM2014/samplesizeANOVA.xlsx`

[6] When $m = 2$, one-way ANOVA is equivalent to the unpaired t-test.

Post	意大利禁区里老是八个人...太夸张了吧 There are always 8 Italian players in their own restricted area…Unbelievable!	**Related Criteria**	**Labels**
Comment 1	我是意大利队的球迷，等待比赛开始。 I am a big fan of the Italian team, waiting for the football match to start.	Coherent	L0
Comment 2	意大利的食物太美味了 Italian food is absolutely delicious.	Topically relevant	L0
Comment 3	太夸张了吧！ Unbelievable!	Non-repetitive	L1
Comment 4	哈哈哈仍然是0：0。还没看到进球。 Haha, it is still 0:0, no goal so far.	Context-independent	L1
Comment 5	这正是意大利式防守足球。 This is exactly the Italian defense style football game.	———	L2

Figure 3: Example of a post and its five candidate comments with human annotation. The content of the post implies that the football match has already started, while the author of Comment 1 is still waiting for the match to start. Comment 2 talks about the food of Italy. Comment 3 is a widely used response, but it is appropriate for this post. Comment 4 states that the current score is still 0:0, it is an appropriate comment only for this specific scenario.

If either (1) or (2) is untrue, the retrieved comment should be labeled "L0"; if either (3) or (4) is untrue, the label should be "L1"; otherwise, the label is "L2".

Figure 3 shows an example of the labeling results of a post and its comments. The first two comments are labeled "L0" because of coherence and topic relevance problems. Comment 3 just repeats the same words of the post, although it is still a comment that the author of the post wanted to see. Comment 4 depends on the context information that the current score is 0:0 and is therefore annotated "L1". Comment 5 is coherent to the post and provides some new and useful information to the author of the post, so it is labeled "L2".

2.4 Chinese Run Results

`NTCIREVAL` was run with options `-g 1:3` (giving a gain value of 1 to each L1-relevant document and 3 to each L2-relevant document) and `-cutoffs 1,10` (output evaluation measures for cutoffs $l = 1, 10$). From the output files, nG@1 is obtained as `MSnDCG@0001`; P+ is obtained as `P-plus`; and nERR@10 is obtained as `nERR@0010`. The SYSDESC (system description) field of each run is listed in Table 10 in the Appendix.

Table 5 gives the official results for the NTCIR-12 STC Chinese Subtask. The runs were sorted by Mean nG@1, P+, and nERR@10. We used a randomized Tukey HSD (honest significant difference) test [2, 11] with $B = 5000$ trials for each evaluation measure; of the $44 * 43/2 = 946$ run pairs, we obtained 192 significant differences with Mean nG@1, 343 significant differences with P+, and 355 significant differences with nERR@10 at the significance level of $\alpha = 0.05$.

Table 6 compares the rankings according to the three evaluation measures in terms of Kendall's τ, with 95% Confidence Intervals (CI's). As the upper limit of each CI is above 1, the three rankings can be considered statistically equivalent.

Here, we provide a brief summary of the results by focusing on the best run from each team for each evaluation measure. In what follows, "$X > Y$" means "X statistically significantly outperforms Y at $\alpha = 0.05$," and the best sets of runs from the statistical point of view are indicated in bold:

- For Mean nG@1,
  BUPTTeam-C-R4 > HITSZ-C-R1, KGO-C-R2, WUST-C-R1;
  MSRSC-C-R1 > HITSZ-C-R1, KGO-C-R2, WUST-C-R1;
  OKSAT-C-R1 > HITSZ-C-R1, KGO-C-R2, WUST-C-R1;
  ITNLP-C-R3 > HITSZ-C-R1, KGO-C-R2, WUST-C-R1;
  splab-C-R1 > HITSZ-C-R1, KGO-C-R2, WUST-C-R1;
  USTC-C-R5 > HITSZ-C-R1, KGO-C-R2, WUST-C-R1;
  uwnlp-C-R2 > KGO-C-R2, WUST-C-R1;
  ICL00-C-R1 > KGO-C-R2, WUST-C-R1;
  Nders-C-R1 > WUST-C-R1;

- For Mean P+,
  BUPTTeam-C-R4 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  MSRSC-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  splab-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  OKSAT-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  USTC-C-R5 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  ITNLP-C-R3 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  ICL00-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  Nders-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  uwnlp-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  cyut-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  PolyU-C-R2 > HITSZ-C-R1, WUST-C-R1;
  Grad1-C-R1 > WUST-C-R1;
  picl-C-R1 > WUST-C-R1;

- For Mean nERR@10,
  BUPTTeam-C-R4 > picl-C-R1, KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  MSRSC-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  splab-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
  Nders-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;

Table 5: Official STC results for the 44 Chinese runs from 16 teams.

Run	Mean nG@1	Run	Mean P+	Run	Mean nERR@10
BUPTTeam-C-R4	0.3567	BUPTTeam-C-R4	0.5082	BUPTTeam-C-R4	0.4945
BUPTTeam-C-R3	0.3533	BUPTTeam-C-R2	0.4933	BUPTTeam-C-R2	0.4830
BUPTTeam-C-R2	0.3533	BUPTTeam-C-R1	0.4883	BUPTTeam-C-R3	0.4805
BUPTTeam-C-R5	0.3467	MSRSC-C-R1	0.4854	BUPTTeam-C-R5	0.4800
BUPTTeam-C-R1	0.3400	BUPTTeam-C-R3	0.4853	BUPTTeam-C-R1	0.4770
MSRSC-C-R1	0.3367	BUPTTeam-C-R5	0.4840	MSRSC-C-R1	0.4592
OKSAT-C-R1	0.3267	splab-C-R1	0.4735	splab-C-R1	0.4449
ITNLP-C-R3	0.3067	OKSAT-C-R1	0.4691	Nders-C-R1	0.4196
splab-C-R1	0.2933	USTC-C-R5	0.4509	ITNLP-C-R3	0.4186
ITNLP-C-R2	0.2900	USTC-C-R1	0.4499	USTC-C-R4	0.4181
USTC-C-R5	0.2867	USTC-C-R4	0.4479	USTC-C-R1	0.4169
uwnlp-C-R2	0.2767	ITNLP-C-R3	0.4445	USTC-C-R5	0.4160
uwnlp-C-R1	0.2767	ICL00-C-R1	0.4359	ITNLP-C-R2	0.4123
USTC-C-R4	0.2767	Nders-C-R1	0.4332	uwnlp-C-R1	0.4095
USTC-C-R1	0.2733	ITNLP-C-R2	0.4320	ICL00-C-R1	0.4066
OKSAT-C-R5	0.2733	USTC-C-R2	0.4310	USTC-C-R2	0.4001
MSRSC-C-R2	0.2733	uwnlp-C-R1	0.4284	OKSAT-C-R1	0.3858
ICL00-C-R1	0.2633	MSRSC-C-R2	0.4208	MSRSC-C-R2	0.3857
USTC-C-R2	0.2567	USTC-C-R3	0.4094	USTC-C-R3	0.3848
OKSAT-C-R3	0.2567	uwnlp-C-R2	0.3977	OKSAT-C-R3	0.3745
OKSAT-C-R2	0.2567	OKSAT-C-R2	0.3976	OKSAT-C-R2	0.3743
Nders-C-R1	0.2533	OKSAT-C-R3	0.3965	uwnlp-C-R2	0.3740
USTC-C-R3	0.2267	cyut-C-R1	0.3851	OKSAT-C-R5	0.3672
cyut-C-R1	0.2233	OKSAT-C-R5	0.3796	cyut-C-R1	0.3608
Grad1-C-R1	0.2100	PolyU-C-R2	0.3603	PolyU-C-R2	0.3426
PolyU-C-R1	0.1900	Grad1-C-R1	0.3585	Grad1-C-R1	0.3361
PolyU-C-R2	0.1867	PolyU-C-R1	0.3510	PolyU-C-R1	0.3314
uwnlp-C-R3	0.1733	picl-C-R1	0.3458	picl-C-R1	0.3196
picl-C-R2	0.1733	picl-C-R2	0.3245	picl-C-R2	0.2985
PolyU-C-R3	0.1667	PolyU-C-R3	0.2968	PolyU-C-R3	0.2771
picl-C-R1	0.1600	OKSAT-C-R4	0.2705	OKSAT-C-R4	0.2488
OKSAT-C-R4	0.1433	uwnlp-C-R3	0.2564	ITNLP-C-R1	0.2354
uwnlp-C-R5	0.1067	ITNLP-C-R1	0.2495	uwnlp-C-R3	0.2255
uwnlp-C-R4	0.1033	MSRSC-C-R3	0.2420	MSRSC-C-R3	0.2236
ITNLP-C-R1	0.1033	uwnlp-C-R4	0.2085	uwnlp-C-R4	0.1867
splab-C-R3	0.0967	splab-C-R2	0.2069	splab-C-R2	0.1831
splab-C-R2	0.0967	KGO-C-R2	0.1926	uwnlp-C-R5	0.1732
MSRSC-C-R3	0.0933	splab-C-R3	0.1896	KGO-C-R2	0.1653
HITSZ-C-R1	0.0933	HITSZ-C-R1	0.1882	splab-C-R3	0.1650
KGO-C-R2	0.0733	uwnlp-C-R5	0.1862	HITSZ-C-R1	0.1544
KGO-C-R1	0.0733	KGO-C-R1	0.1480	KGO-C-R1	0.1281
WUST-C-R1	0.0567	WUST-C-R1	0.1218	WUST-C-R1	0.0980
HITSZ-C-R3	0.0400	HITSZ-C-R2	0.0856	HITSZ-C-R2	0.0725
HITSZ-C-R2	0.0133	HITSZ-C-R3	0.0836	HITSZ-C-R3	0.0701

Table 6: Run ranking similarity across three measures: Kendall's τ values with 95% CIs.

	Mean nG@1	P+
P+	.854 [.649, 1.059]	-
nERR@10	.848 [.643, 1.053]	.926 [.721, 1.131]

ITNLP-C-R3 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
USTC-C-R4 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
uwnlp-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
ICL00-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
OKSAT-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
cyut-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
PolyU-C-R2 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
Grad1-C-R1 > KGO-C-R2, HITSZ-C-R1, WUST-C-R1;
picl-C-R1 > HITSZ-C-R1, WUST-C-R1;

3. JAPANESE SUBTASK

This section describes the details of the Japanese subtask as well as the results of the submitted runs. We had seven participating teams. Each team was allowed to submit up to five runs. We received 25 runs in total (See Fig.7).

Table 7: Organization and number of submitted runs of participating groups in the STC Japanese subtask.

Group ID	Organization	#runs
KIT15	Kyoto Institute of Technology	4
NTTCS	NTT Communication Science Labs.	2
OKSAT	Osaka Kyoiku University	5
Oni	Osaka University	5
SLSTC	Waseda University	3
sss	University of Tokyo	2
yuila	Yamagata University	4

3.1 Task Definition

The task definition is basically the same as that for the Chinese subtask, although in the Japanese subtask, we use Twitter instead of Weibo. The task definition is summarized below.

- In the **training** phrase, participants are provided with train-

ing data (pairs of tweets) labeled with relevance labels L0, L1, or L2. Participants develop their own models to retrieve comments for a given tweet.

- In the **test** phase, participants are given a set of test tweets. Each system outputs a ranked list of up to ten tweets as a comment for a given tweet.
- In the **evaluation** phase, all the results are pooled and labeled by humans.

3.2 Evaluation Measures

We adopted $nG@1$ and $nERR@5$ (only the top five comments were evaluated due to budget reasons) as evaluation measures, which are described in Section 3.2. However, an important difference from the original definition is that since multiple relevance labels are used (See Section3.4) for each comment, we define $g(r)$ as *averaged gain* as follows:

$$g(r) = \frac{\sum_{i=1}^{n} g_i(r)}{n},$$

where n is the number of labels given to each comment (in our setting, $n = 10$), and $g_i(r)$ is an i-th relevance label for the comment at rank r. With this averaged gain, we can use the same definition of $nG@1$ and $nERR@5$ as in the Chinese task. P^+ was not used in the Japanese task because it is not trivial to define this measure on averaged gains.

3.2.1 $Acc_G@k$

In addition to $nG@1$ and $nERR@5$, we use *accuracy* $Acc_G@k$:

$$Acc_G@k = \frac{1}{nk} \sum_{r=1}^{k} \sum_{i=1}^{n} \delta(l_i(r) \in G),$$

where $l_i(r)$ is an i-th relevance label. G specifies relevance labels regarded as "correct". This measure computes the averaged counts of the number of labels judged as correct ($l_i(r) \in G$). In this task, we evaluated the results with $G = \{L2\}$ and $G = \{L1, L2\}$, for $k = 1$ and $k = 5$.

3.3 Japanese Test Collection

We created the Japanese test collection by crawling Twitter. This section describes how we created the Twitter corpus as well as the training and test data.

All the data are provided in GitHub.[7] However, due to a license issue, we provided only tweet IDs instead of raw text. Participants were requested to crawl the tweet texts by themselves or to purchase tweet data from NTT Data, who is the sole selling agency for Twitter data in Japan.

3.3.1 The Twitter Corpus

In order to reduce noise and obtain tweets that covered events during an entire year, the following procedure was used to create the Twitter corpus. This process was done with the help of NTT Data, which had full access to the Twitter data.

1. Tweets (from 1st January 2014 to 31st December 2014) that matched the following conditions were extracted. The extracted tweets are referred to as **T1**.
   (a) The tweet has an in_reply_to_status_id_str field.
   (b) The tweet is not a retweeted tweet.
   (c) The tweet has more than 20 characters.
   (d) The tweet consists only of English letters, numbers, Japanese characters, punctuation marks, and white spaces.
   (e) The tweet does not have URLs indicated by http:// or https://.
   (f) The tweet's screen_name does not contain "bot".
   (g) The tweet does not consist only of ascii characters.
2. Tweets (also from 1st January 2014 to 31st November 2014) that match the following conditions were extracted. The extracted tweets are referred to as **T2**.
   (a) The tweet does *not* have an in_reply_to_status_id_str field; that is, it is an initial post.
   (b) The tweet matches the conditions 1b–1g used to extract T1.
3. Just over 0.5M tweets from T1 were randomly sampled, and the pool of tweets was initialized.
4. For each sampled tweet, using its in_reply_to_status_id_str field, a tweet from T2 was searched for to make a pair. After making sure that the IDs (id_str) of the paired tweets were unique within a pool, it was added to the pool. When the number of tweets in the pool reached 1M, this procedure was terminated.

3.3.2 Training Data

We created our training data in the following manner. First, we randomly sampled 200 tweets from 1st January 2015 to 30th June 2015. Here, the tweets were those that satisfied the conditions for T2 in Section 3.3.1. Then, for each sampled tweet, we retrieved up to ten tweets from the Twitter corpus.

For this retrieval, we indexed the Twitter corpus with Lucene (version 5.2.1 was used) using the built-in JapaneseAnalyzer. Here, a document to be added to the index was a pair of tweets; a document has fields t1 and t2, corresponding respectively to the tweets from T1 and T2. Given an input tweet, the index was searched with t2 as the target field using Lucene's default search parameters, and the top five documents were retrieved. We used the t1 and t2 fields of the top five documents for relevance assessment.

We used crowdsourcing for relevance assessment. Each retrieved tweet was labeled with L0, L1, or L2 by ten annotators (crowd workers). Some tweets were labeled 'NA', which indicates that an annotator judged the original post (input tweet) as meaningless. The training data consisted of 1,959 pairs of tweets (input tweets coupled with retrieved tweets).

Figure 4 shows an example post and its three candidate comments with human annotations.

3.3.3 Test Data

In order to create the test data, we randomly sampled 250 tweets from 1st January 2015 to 30th June 2015. This is a disjoint set from the training data. The tweets satisfy the conditions for T2 in Section 3.3.1. We also made it sure that these tweets are not meaningless by preliminary annotation. The tweets were distributed to the participants of the Japanese subtask. Note that due to the nature of Twitter, where tweets are deleted at the discretion of users, we used 202 tweets as test data. In addition, we did not evaluate retrieved tweets that did not exist at the time of the relevance assessment.

[7] https://github.com/mynlp/stc

Post	ああ一次の日曜日お好み焼き食べたいって言われてた気がする Ah, someone told me he wants to eat Okonomi-yaki this Sunday.	**Labels**
Comment 1	週末とか代々木とかでフェスやってるんじゃね？ Some festival will be held in Yoyogi this weekend, maybe?	0001010101
Comment 2	屋台のお好み焼きが食べたい･･･どっかで縁日してないかなぁ･･･ I wanna eat Okonomi-yaki in a stall... I wanna join a festival somewhere...	0112122020
Comment 3	お好み焼きが食べたい！だれか今度みんなでいこう！てかおいしいお好み焼き屋知ってる人！ I wanna eat Okonomi-yaki! Anybody want to join me? Does anyone know a good Okonomi-yaki restaurant?	2202211222

Figure 4: An example post and its three candidate comments with human annotation. The numbers 0, 1 and 2 indicate relevance labels L0, L1, and L2, respectively.

Table 8: Official STC results for the 25 Japanese runs from 7 teams.

Run	Mean $nG@1$	Run	Mean $nERR@5$	Run	Mean $Acc_{L2}@1$
OKSAT-J-R1	0.6794	OKSAT-J-R1	0.7805	OKSAT-J-R1	0.4574
OKSAT-J-R2	0.6756	OKSAT-J-R2	0.7754	OKSAT-J-R2	0.4520
KIT15-J-R3	0.4112	KIT15-J-R3	0.5332	KIT15-J-R3	0.2297
OKSAT-J-R5	0.4014	OKSAT-J-R5	0.5000	KIT15-J-R1	0.1817
KIT15-J-R1	0.3345	KIT15-J-R1	0.4573	sss-J-R1	0.1817
KIT15-J-R2	0.3273	KIT15-J-R2	0.4425	KIT15-J-R2	0.1812
sss-J-R1	0.2837	Oni-J-R1	0.3961	OKSAT-J-R5	0.1807
OKSAT-J-R3	0.2713	Oni-J-R2	0.3886	yuila-J-R1	0.1470
OKSAT-J-R4	0.2499	OKSAT-J-R3	0.3825	OKSAT-J-R3	0.1460
yuila-J-R1	0.2446	Oni-J-R3	0.3763	OKSAT-J-R4	0.1361
Oni-J-R1	0.2266	sss-J-R1	0.3711	Oni-J-R1	0.1198
Oni-J-R2	0.2192	OKSAT-J-R4	0.3620	Oni-J-R2	0.1114
SLSTC-J-R2	0.2091	Oni-J-R5	0.3385	Oni-J-R3	0.1084
Oni-J-R3	0.2036	yuila-J-R1	0.3278	Oni-J-R5	0.1010
Oni-J-R5	0.1911	Oni-J-R4	0.3265	NTTCS-J-R1	0.0921
NTTCS-J-R2	0.1849	NTTCS-J-R2	0.2481	Oni-J-R4	0.0891
Oni-J-R4	0.1827	NTTCS-J-R1	0.2383	NTTCS-J-R2	0.0876
NTTCS-J-R1	0.1700	SLSTC-J-R2	0.2325	KIT15-J-R4	0.0787
yuila-J-R4	0.1561	yuila-J-R4	0.2288	SLSTC-J-R2	0.0782
yuila-J-R2	0.1549	yuila-J-R3	0.2284	yuila-J-R4	0.0663
yuila-J-R3	0.1549	yuila-J-R2	0.2276	yuila-J-R2	0.0649
KIT15-J-R4	0.1450	KIT15-J-R4	0.2224	yuila-J-R3	0.0649
sss-J-R2	0.1081	sss-J-R2	0.1804	sss-J-R2	0.0634
SLSTC-J-R1	0.0997	SLSTC-J-R1	0.1671	SLSTC-J-R1	0.0381
SLSTC-J-R3	0.0201	SLSTC-J-R3	0.0256	SLSTC-J-R3	0.0054

Run	Mean $Acc_{L2}@5$	Run	Mean $Acc_{L1,L2}@1$	Run	Mean $Acc_{L1,L2}@5$
OKSAT-J-R1	0.3673	OKSAT-J-R1	0.7817	OKSAT-J-R1	0.7050
OKSAT-J-R2	0.3583	OKSAT-J-R2	0.7807	OKSAT-J-R2	0.6865
KIT15-J-R3	0.2050	OKSAT-J-R5	0.5965	KIT15-J-R3	0.5380
KIT15-J-R1	0.1743	KIT15-J-R3	0.5589	OKSAT-J-R5	0.5196
sss-J-R1	0.1730	KIT15-J-R1	0.4748	KIT15-J-R1	0.4535
KIT15-J-R2	0.1660	KIT15-J-R2	0.4614	KIT15-J-R2	0.4317
OKSAT-J-R3	0.1458	OKSAT-J-R3	0.3876	Oni-J-R1	0.3910
Oni-J-R1	0.1444	sss-J-R1	0.3797	Oni-J-R3	0.3887
Oni-J-R3	0.1390	OKSAT-J-R4	0.3574	Oni-J-R2	0.3742
Oni-J-R2	0.1376	yuila-J-R1	0.3480	OKSAT-J-R3	0.3683
OKSAT-J-R4	0.1366	SLSTC-J-R2	0.3416	OKSAT-J-R4	0.3543
OKSAT-J-R5	0.1282	Oni-J-R1	0.3416	sss-J-R1	0.3495
yuila-J-R1	0.1267	Oni-J-R2	0.3381	Oni-J-R5	0.3454
Oni-J-R5	0.1248	Oni-J-R3	0.2955	Oni-J-R4	0.3329
Oni-J-R4	0.1106	NTTCS-J-R2	0.2946	yuila-J-R1	0.3087
sss-J-R2	0.0776	Oni-J-R4	0.2807	NTTCS-J-R2	0.2333
KIT15-J-R4	0.0720	Oni-J-R5	0.2703	NTTCS-J-R1	0.2318
NTTCS-J-R1	0.0698	NTTCS-J-R1	0.2639	yuila-J-R4	0.2254
NTTCS-J-R2	0.0677	yuila-J-R4	0.2490	yuila-J-R2	0.2254
yuila-J-R4	0.0568	yuila-J-R2	0.2485	yuila-J-R3	0.2254
yuila-J-R3	0.0568	yuila-J-R3	0.2485	KIT15-J-R4	0.2130
yuila-J-R2	0.0567	KIT15-J-R4	0.2114	sss-J-R2	0.1823
SLSTC-J-R1	0.0364	SLSTC-J-R1	0.1644	SLSTC-J-R2	0.1795
SLSTC-J-R2	0.0332	sss-J-R2	0.1609	SLSTC-J-R1	0.1650
SLSTC-J-R3	0.0032	SLSTC-J-R3	0.0391	SLSTC-J-R3	0.0196

3.4 Relevance Assessments

For each tweet, up to ten results were allowed. However, for budget reasons, we used only the top five retrieved tweets for relevance assessment. All the retrieved tweets from the participating teams were labeled L0, L1, or L2. See the definitions for the labels in Section 2.3.4.

For labeling each retrieved tweet, since the labeling task can be quite subjective, we used ten annotators. The inter-annotator agreement of the relevance labels in Fleiss' κ was rather low at 0.317, confirming the subjective nature of the task. When we merge L1 and L2 and make it a two-class annotation, the agreement becomes 0.421 (moderate agreement), showing that the annotators share some common conception about the relevance in short text conversation. Here, the κ is similar to that of a similar task in dialogue research [4, 5].

3.5 Japanese Run Results

Table 8 lists the official STC results for the 25 Japanese runs from 7 teams. The descriptions of the runs are given in Table 11 in the Appendix. The runs have been sorted by the mean values of the evaluation measures. As indicated in the table, OKSAT outperformed the other teams in all metrics. OSKAT is followed by KIT15.

Following the Chinese subtask, we also used a randomized Tukey HSD test [2, 11] with $B = 1000$ trials for each evaluation measure; of the $25 * 24/2 = 300$ run pairs, we obtained 159 significant differences with Mean $nG@1$, 205 significant differences with Mean $nERR@5$, 140 significant differences with Mean $Acc_{L2}@1$, 188 significant differences with Mean $Acc_{L2}@5$, 173 significant differences with Mean $Acc_{L1,L2}@1$, and 210 significant differences with Mean $Acc_{L1,L2}@5$ at the significance level of $\alpha = 0.05$.

Table 9 compares the rankings according to the six evaluation measures in terms of Kendall's τ, with 95% confidence intervals.

We provide a brief summary of the results by focusing on the best run from each team for each evaluation measure. In what follows, "$X > Y$" means "X statistically significantly outperforms Y at $\alpha = 0.05$", and the best sets of runs from a statistical point of view are indicated in bold:

- For Mean $nG@1$,
  OKSAT-J-R1 > KIT15-J-R3, sss-J-R1, yuila-J-R1, Oni-J-R1, SLSTC-J-R2, NTTCS-J-R2;
  KIT15-J-R3 > sss-J-R1, yuila-J-R1, Oni-J-R1, SLSTC-J-R2, NTTCS-J-R2;
  sss-J-R1 > NTTCS-J-R2;

- For Mean $nERR@5$,
  OKSAT-J-R1 > KIT15-J-R3, Oni-J-R1, sss-J-R1, yuila-J-R1, NTTCS-J-R2, SLSTC-J-R2;
  KIT15-J-R3 > Oni-J-R1, sss-J-R1, yuila-J-R1, NTTCS-J-R2, SLSTC-J-R2;
  Oni-J-R1 > NTTCS-J-R2, SLSTC-J-R2;
  sss-J-R1 > NTTCS-J-R2, SLSTC-J-R2;
  yuila-J-R1 > SLSTC-J-R2;

- For Mean $Acc_{L2}@1$,
  OKSAT-J-R1 > KIT15-J-R3, sss-J-R1, yuila-J-R1, Oni-J-R1, NTTCS-J-R1, SLSTC-J-R2;
  KIT15-J-R3 > yuila-J-R1, Oni-J-R1, NTTCS-J-R1, SLSTC-J-R2;
  sss-J-R1 > NTTCS-J-R1, SLSTC-J-R2;

- For Mean $Acc_{L2}@5$,
  OKSAT-J-R1 > KIT15-J-R3, sss-J-R1, Oni-J-R1, yuila-J-R1, NTTCS-J-R1, SLSTC-J-R1;
  KIT15-J-R3 > Oni-J-R1, yuila-J-R1, NTTCS-J-R1, SLSTC-J-R1;
  sss-J-R1 > NTTCS-J-R1, SLSTC-J-R1;
  Oni-J-R1 > NTTCS-J-R1, SLSTC-J-R1;
  yuila-J-R1 > NTTCS-J-R1, SLSTC-J-R1;

- For Mean $Acc_{L1,L2}@1$,
  OKSAT-J-R1 > KIT15-J-R3, sss-J-R1, yuila-J-R1, Oni-J-R1, SLSTC-J-R2, NTTCS-J-R2;
  KIT15-J-R3 > sss-J-R1, yuila-J-R1, Oni-J-R1, SLSTC-J-R2, NTTCS-J-R2;

- For Mean $Acc_{L1,L2}@5$,
  OKSAT-J-R1 > KIT15-J-R3, Oni-J-R1, sss-J-R1, yuila-J-R1, NTTCS-J-R2, SLSTC-J-R2;
  KIT15-J-R3 > Oni-J-R1, sss-J-R1, yuila-J-R1, NTTCS-J-R2, SLSTC-J-R2;
  Oni-J-R1 > NTTCS-J-R2, SLSTC-J-R2;
  sss-J-R1 > NTTCS-J-R2, SLSTC-J-R2;
  yuila-J-R1 > SLSTC-J-R2;

4. CONCLUSIONS AND FUTURE WORK

This paper presented an overview of the Short Text Conversation (STC) pilot task at NTCIR-12. This task aims to build a conversation system by maintaining a large repository of post-comment pairs, then finding a sophisticated way to reuse these existing comments to respond to new posts. At NTCIR-12, STC consists of two subtasks: the Chinese subtask, and the Japanese subtask. The main difference between the two subtasks lies in the sources and languages of the post-comment repository.

4.1 Chinese Subtask

For the Chinese subtask, we drew the following conclusions from a brief analysis of the methods used by twelve teams:

- Filtering comments by using some manually designed rules was simple but effective. Two of the top three teams used this strategy.
- Representing a post (or comment) by the word2vec model [9] was helpful to perform semantic-level matching. Of the top six teams, half of them used word2vec representations.
- It was interesting to see that the Group ITNLP modeled STC as a "representation learning" problem, in which they first extracted a lot of local and trivial matching patterns, then used deep learning models to obtain effective high-level patterns. They achieved fourth place.

In future runs, participants will be able to:

- Perform more analysis on the properties of post-comment pairs from the aspects of comment length, popularity, dialogue act, and sentiment to obtain more effective filtering rules.
- Pay more attention to the "representation learning" method in order to automatically learn high-level and effective patterns. Determining how to effectively combine these learned features with generally used features (e.g. the vector space model) will also be an interesting research problem.

Table 9: Run ranking similarity across the six measures: Kendall's τ values with 95% CIs.

	nG@1	$nERR$@5	Acc_{L2}@1	Acc_{L2}@5	$Acc_{L1,L2}$@1	$Acc_{L1,L2}$@5
nG@1	-	.871 [.747, .996]	.878 [.768, .988]	.729 [.538, .92]	.958 [.909, 1.007]	.809 [.638, .98]
$nERR$@5	.871 [.747, .996]	-	.829 [.694, .965]	.765 [.595, .934]	.843 [.708, .978]	.935 [.853, 1.017]
Acc_{L2}@1	.878 [.768, .988]	.829 [.694, .965]	-	.807 [.663, .952]	.836 [.71, .961]	.780 [.626, .934]
Acc_{L2}@5	.729 [.538, .92]	.765 [.595, .934]	.807 [.663, .952]	-	.687 [.484, .89]	.782 [.625, .938]
$Acc_{L1,L2}$@1	.958 [.909, 1.007]	.843 [.708, .978]	.836 [.71, .961]	.687 [.484, .89]	-	.780 [.6, .96]
$Acc_{L1,L2}$@5	.809 [.638, .98]	.935 [.853, 1.017]	.780 [.626, .934]	.782 [.625, .938]	.780 [.6, .96]	-

- The computation of deep matching models is very time-consuming, so finding an effective means of model compression will also be an important research topic.

4.2 Japanese Subtask

For the Japanese subtask, we drew the following conclusions from a brief analysis of the methods used by seven teams:

- A simple application of neural networks (NNs) did not lead to good results. Heuristic rules and similarity based methods performed better than machine learning based methods. This is probably due to the difficulty of learning reasonable models from the small number of training examples.
- Clustering of utterances seems to be a reasonable way to abstract tweets. The second- and third-placed teams used clustering. Clustering can be combined with NN-based methods to effectively learn the utterance space (as indicated by the results of Team sss);
- The effectiveness of external dialogue data was limited; this was probably because of the differences between tweets and ordinary dialogues. Team sss, who used external tweets, performed better than those who used external dialogue data.

In the future, participants can aim to abstract utterances (tweets) in order to reduce the search space and to make it possible for NN-based methods to work. We also consider that for STC to be truly useful for dialogue systems, we need to consider differences between CGMs and dialogue data.

4.3 Concluding Remarks

We summarize the results in both subtasks. For both subtasks, it is interesting to see that some heuristic or manually designed rules based on comprehensive analysis (e.g. by a clustering algorithm) on the properties of post-comment pairs tended to achieve better performance than simple application of some sophisticated models. Additionally, the recent NN-based models also showed their potential in automatic learning from trivial local matching features to perform competitively. In future runs, participants will be encouraged to focus more on striking a better balance between manually designing features and automatically learning features. STC is the largest task of NTCIR-12, so we plan to continue to run this task at NTCIR-13 and look forward to seeing new improvements at the next round.

5. ACKNOWLEDGMENTS

We would like to thank all the STC task participants for their effort in exploring new techniques and submitting their runs and reports. We also thank general chairs and program co-chairs of NTCIR-12 for their encouragement and support.

6. REFERENCES

[1] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[2] B. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM TOIS*, 30(1), 2012.

[3] O. Chapelle, S. Ji, C. Liao, E. Velipasaoglu, L. Lai, and S.-L. Wu. Intent-based diversification of web search results: Metrics and algorithms. *Information Retrieval*, 14(6):572–592, 2011.

[4] R. Higashinaka, K. Funakoshi, M. Araki, H. Tsukahara, Y. Kobayashi, and M. Mizukami. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of SIGDIAL*, pages 87–95, 2015.

[5] R. Higashinaka, M. Mizukami, K. Funakoshi, M. Araki, H. Tsukahara, and Y. Kobayashi. Fatal or not? finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proceedings of EMNLP*, pages 2243–2248, 2015.

[6] D. Ishikawa, T. Sakai, and N. Kando. Overview of the ntcir-8 community qa pilot task (part i): The test collection and the task. *NTCIR-8 proceedings*, pages 421–432, 2010.

[7] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 20(4):422–446, 2002.

[8] J. Lin and M. Efron. Overview of the trec-2013 microblog track. In *Proceeddings of the 22th Text REtrieval Conference (TREC 2013)*, 2013.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[10] T. Sakai. Bootstrap-based comparisons of IR metrics for finding one relevant document. In *AIRS 2006 (LNCS 4182)*, pages 374–389, 2006.

[11] T. Sakai. Metrics, statistics, tests. In *PROMISE Winter School 2013: Bridging between Information Retrieval and Databases (LNCS 8173)*, 2014.

[12] T. Sakai. *Information Access Evaluation Methodology: For the Progress of Search Engines (in Japanese)*. Coronasha, 2015.

[13] T. Sakai. Topic set size design. *Information Retrieval Journal*, 2015.

[14] T. Sakai and S. Robertson. Modelling a user population for designing information retrieval metrics. In *Proceedings of EVIA 2008*, pages 30–41, 2008.

[15] T. Sakai, L. Shang, Z. Lu, and H. Li. Topic set size design with the evaluation measures for short text conversation. In *Proceedings of AIRS 2015 (LNCS 9460)*, 2015.

[16] L. Shang, Z. Lu, and H. Li. Neural responding machine for short-text conversation. In *Proceedings of ACL 2015*, pages 1577–1586, 2015.

[17] H. Wang, Z. Lu, H. Li, and E. Chen. A dataset for research on short-text conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 935–945, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

APPENDIX

Table 10: SYSDESC fields of the 44 Chinese runs. Note that not all of them are informative.

BUPTTeam-C-R1:	[post+cmnt]
BUPTTeam-C-R2:	[post+cmnt+random walk based similary]
BUPTTeam-C-R3:	[post+cmnt+random walk based similary+rank based R1]
BUPTTeam-C-R4:	[post+cmnt+pagerank based similary+time rule]
BUPTTeam-C-R5:	[post+cmnt+pagerank based similary+time rule+rank with R1]
Grad1-C-R1:	2015-11-26-Gard1-C-R1
HITSZ-C-R1:	Learning to rank post-comment pairs based convolutional neural networks (CNN) model, which trained by using Pairwise strategy.
HITSZ-C-R2:	Learning to rank post-comment pairs based convolutional neural networks (CNN) model, which trained by using Pointwise strategy.
HITSZ-C-R3:	baseline: computing cosine-similarity between post and comment to rank the results
ICL00-C-R1:	Our method applies a rich-feature model using semantic, grammar, n-gram and string features to extract high-level semantic meanings of text.
ITNLP-C-R1:	[DNNPatternsearly_stopping_1.2million]
ITNLP-C-R2:	[LRPatternsL15Negtive]
ITNLP-C-R3:	[LRPatternsL19Negtive]
KGO-C-R1:	[insert a short description in English here]
KGO-C-R2:	LambdaMART using features extracted through deep learning
MSRSC-C-R1:	[MSRSCCR1]
MSRSC-C-R2:	[MSRSCCR2]
MSRSC-C-R3:	[MSRSCCR3]
Nders-C-R1:	[A run file with our algorithm]
OKSAT-C-R1:	Gram Base Index, Probabilistic Model
OKSAT-C-R2:	Gram Base Index, Probabilistic Model
OKSAT-C-R3:	Gram Base Index, Probabilistic Model
OKSAT-C-R4:	Gram Base Index, Probabilistic Model
OKSAT-C-R5:	Gram Base Index, Probabilistic Model
PolyU-C-R1:	baseline method + 3 features + Keyword expansion_2
PolyU-C-R2:	baseline method + 3 features + Keyword expansion method_1
PolyU-C-R3:	baseline method
USTC-C-R1:	system : tfidfqpqr + encDecforwardreverse + transitionP2c
USTC-C-R2:	system : tfidfqpqr + encDecforwardreverse + jointTrain
USTC-C-R3:	system : tfidfqpqr + encDecforward + transitionP2c
USTC-C-R4:	system : tfidfqpqr + encDecforwardreverse
USTC-C-R5:	baseline system : tfidfqpqr + encDecforward
WUST-C-R1:	We propose formalizing short text conversation as a search problem at the first step, and employing stateofthe-art information retrieval (IR) techniques to carry out the task. The system performs retrieval-based short text conversation in three-stages, they are retrieval, matching and ranking. In the first stage (retrieval), we remove punctuation marks and emotions, and use ICTCLAS for Chinese word segmentation. We use the basic linear matching models for the second stage (matching).
cyut-C-R1:	[The system will be A repository of post-comment pairs from Sina Weibo used for training, use jcseg be hyphenation, indexing, and then test topics to use lucene search results in response to the sentence.]
picl-C-R1:	This run first tries to find the most similar posts in the repository to the posts in the test set and use their most similar comments to the post in the test set as answers.
picl-C-R2:	This run first tries to find the most similar posts in the repository to the posts in the test set and use comments as answers.
splab-C-R1:	3
splab-C-R2:	CDSSM
splab-C-R3:	CDSSM
uwnlp-C-R1:	BF ranking, max{sim(query, post) + sim(query, comment)}
uwnlp-C-R2:	BF ranking, max sim(query, post) first, then max{sim(query, comment)
uwnlp-C-R3:	BF ranking, max sim(query, post) first, then longest comments
uwnlp-C-R4:	ML ranking, max{sim(query, post) + sim(query, comment)}
uwnlp-C-R5:	ML ranking, max sim(query, post) first, then max{sim(query, comment)}.

Table 11: SYSDESC fields of the 25 Japanese runs.

KIT15-J-R1:	The ratio of the weight of the idf and LDA in the semantic similarity is 6:4.
KIT15-J-R2:	The ratio of the weight of the idf and LDA in the semantic similarity is 5:5.
KIT15-J-R3:	Semantic similarity by only idf.
KIT15-J-R4:	Semantic similarity by only LDA.
NTTCS-J-R1:	IR-status based on the word2vec distance (average)
NTTCS-J-R2:	Dialogue breakdown detection (O score) with IR-status for candidates selection
OKSAT-J-R1:	post(full+partial)+cmnt(phrase+word), short sentence, word filter, merge queries
OKSAT-J-R2:	post(full+partial)+cmnt(phrase+word), short sentence, merge multiple queries
OKSAT-J-R3:	difference between mecab and expand mecab, length merge queries.
OKSAT-J-R4:	difference between mecab and expand mecab queries.
OKSAT-J-R5:	This run's cmnt is all short cmnt of agreement
Oni-J-R1:	Random Forest from TF-IDF corpus and cosine similarity
Oni-J-R2:	add score of Random Forest corpus to cosine similarity using TF-IDF
Oni-J-R3:	TF-IDF model and cosine similarity
Oni-J-R4:	word2vec => TF-IDF model and cosine similarity
Oni-J-R5:	Weighted Text Matrix Factorization model
SLSTC-J-R1:	Learning using Error Back Propagation
SLSTC-J-R2:	Using Pagerank for Lexical network
SLSTC-J-R3:	Using Pagerank for Lexical network exclding w characters
sss-J-R1:	The system selects replies based on the perplexities of concatenated tweet-reply pairs in LSTM language model from the 500-best results of R2.
sss-J-R2:	We use our kernel-based classifier that estimates scores of each tweet-reply pair by bag-of-words features.
yuila-J-R1:	Our run1 system selects an output tweet that is the most similar to an input tweet by TF-IDF and cosine similarity.
yuila-J-R2:	We use the chat-dialogue-corpus(https://sites.google.com/site/dialoguebreakdowndetection/chat-dialogue-corpus). First, our run2 system searches an utterance(UttA) that is the most similar to an input tweet by TF-IDF and cosine similarity, from chat-dialogue-corpus. Then, this system uses an utterance(UttB) next to UttA. UttB is a response utterance of UttA. Finally, this system selects an output tweet that is the most similar to UttB by TF-IDF and cosine similarity.
yuila-J-R3:	Our run3 system is a system merged run1 and run2. Top five output tweets are selected by run2. Others are selected by run1.
yuila-J-R4:	Our run4 system is a system merged run1 and run2. If there are same output tweets in tweets selected by run1 and run2, this system ranks them higher. Others are the same as run3.