SLSTC at the NTCIR-12 STC Task

Hiroto Denawa
Waseda University
hi_denawa@fuji.waseda.jp

Tomoaki Sano
Waseda University
tonosamoaki@asagi.waseda.jp

Yuta Kadotami
Waseda University
kdtm-783640@ruri.waseda.jp

Sosuke Kato
Waseda University
sow@suou.waseda.jp

Tetsuya Sakai
Waseda University
tetsuyasakai@acm.org

ABSTRACT

The SLSTC team participated in the NTCIR-12 Short Text Conversation (STC)[1] task. This report describes our approach to solving the STC problem and discusses the official results.

Team Name

SLSTC

Subtasks

Japanese subtask

Keywords

STC; HNN; Word2Vec; PageRank; word co-occurrence network

1. INTRODUCTION

SLSTC (The Sakai Laboratory, Waseda University) participated in the Japanese subtask of the STC task. This paper briefly describes our approaches, and reports on the official results.

Table 1.1 shows the list of runs that we submitted to the STC Japanese subtask. In Section 2, we describe the algorithms we employed to generate these runs. In Section 3, we discuss the official results of our runs. Finally, in Section 4, we conclude this paper and lists up future work items.

Table 1.1: the list of runs

run name
SLSTC-J-R1
SLSTC-J-R2
SLSTC-J-R3

2. METHODS

2.1 SLSTC-J-R1

This run was generated by the second author (Tomoaki Sano), as his bachelor's thesis project.

First, Word2Vec [1] is utilised to generate a distributed representation for every term in a tweet. Japanese Wikipedia and Nicopedia (Niconico Daihyakka) were used as the corpora. Thus every term is represented as a word vector. Second, a tf-idf weight for each term in a tweet and the word vectors are weighted accordingly. The tweet is then represented as the sum of the weighted word vectors. Third, a three-layered neural network is used for generating a reply from a post. Finally, the STC repository is searched and the top five replies that are most similar to the output from the neural network are included in the run file.

[1] http://code.google.com/p/word2vec/

2.2 SLSTC-J-R2 and SLSTC-J-R3

These two runs were generated by the first, third, and the fourth authors as a collaboration project. Our method is based on a word co-occurrence network, and it consists of three parts: network construction, subnetwork extraction, and ranked output generation.

2.2.1 Network Construction

In this part, we make a word co-occurrence network from post-reply tweet pairs. As the dataset, 427,200 post-reply tweet pairs are provided from NTCIR. First, we perform morphological analysis on each tweet in the all tweets using MeCab[3]. Let t be a tweet from the post-reply tweet pair data, and let $m = \{w_1, w_2, \ldots, w_n\}$ be a word sequence obtained by performing morphological analysis on t. Let $M = \{m\}$ be the set of word sequences obtained from all tweets in the tweet pair data. We define the set of nodes for a word co-occurrence network as follows:

$$V = \{w \in m |\ m \in M\}\ . \tag{2.1}$$

Let $B(m)$ denote the set of all word bigrams obtained from m, i.e., $\{< w_1, w_2 >, < w_2, w_3 >, \ldots, < w_{n-1}, w_n >\}$. Let $m_P \in M$ be the word sequence obtained from a post tweet P, and let $m_{R(P)} \in M$ be the word sequence obtained from a reply tweet $R(P)$ that was a response to P. Let $C(m_P, m_{R(P)})$ be the set of all ordered word pairs $\{< w_P, w_{R(P)} >\}$, where w_P is from m_P and $w_{R(P)}$ is from $m_{R(P)}$. The edges of the aforementioned word co-occurrence network as defined as:

$$E = E_{bi} \cup E_{PR} \tag{2.2}$$

where

$$E_{bi} = \bigcup_{m \in M} \{b \in B(m)\}\ , \tag{2.3}$$

$$E_{PR} = \bigcup_{m_P \in M} \{c \in C(m_P, m_{R(P)})\}\ . \tag{2.4}$$

Finally, our word co-occurrence network, built from the tweet pair data, is denoted as:

$$G = (V, E) \ . \qquad (2.5)$$

2.2.2 Subnetwork Extraction

Given a test post tweet i, we extract a subgraph of G, which we denote by G', as follows. Let I be the word sequence obtained by performing morphological analysis on i and removing all words that are not in V. We obtain a set of nodes $V' \subseteq V$ and a set of edges $E' \subseteq E$ as:

$$V' = \{w_I \in I\} \cup \{w| < w_I, w >\in E, \ w_I \in I\} \ , \qquad (2.6)$$

$$E' = \{< w_I, w >\in E | w_I \in I\} \ . \qquad (2.7)$$

Finally, the subgraph for that test tweet is obtained as:

$$G' = (V', E') \ . \qquad (2.8)$$

2.2.3 Ranked Output Generation

We rank all the tweets in the tweet pair repository based on the aforementioned subgraph G' for each given test tweet as follows.

For a given test tweet, we compute the PageRank $PR(w)$ of each node $w \in V'$. Let d be a parameter; we set it to $d = 0.9$ based on a pilot experiment. Let $V'(w) = \{w' \in V' | < w', w >\in E'\}$, that is, the set of nodes with an edge going into w. Also, let $E(w')$ be the set of outgoing edges from node w'. $PR(w)$ is initially set to $PR(w) = \frac{1-d}{|V'|}$ and is updated through 100 iterations as follows:

$$PR(w) = \frac{1-d}{|V'|} + \sum_{w' \in V'(w)} \frac{d * PR(w')}{|E(w')|} \ . \qquad (2.9)$$

Next, for each candidate tweet t in the repository whose word sequence is denoted by m, we compute a tfidf-like score as follows. Let $tf(w, m)$ denote the number of occurrences of w in word sequence m and let $idf(w) = \log_{10} \frac{|M|}{|\{m|w \in m\}|} + 1$. The tfidf-like score for w from m is computed as:

$$tfidf(w, m) = \frac{|m| - tf(w, m) + 1}{|m|} idf(w) \ . \qquad (2.10)$$

The final score for tweet t, whose word sequence is m, is given by:

$$Score(t) = \sum_{w \in W} tfidf(w, m) * PR(w) \ , \qquad (2.11)$$

where W is the set of words from m that are also in V'. The candidate tweets are sorted by this score, and the top ranked tweets are returned as the output.

The only difference between our two runs SLSTC-J-R2 and SLSTC-J-R3 is that the latter removed all continuous occurrences of "w" in the test post tweets, which are often used in Japanese social media in a way similar to the English "lol" (laughing out loud).

3. OFFICIAL RESULTS AND DISCUSSIONS

Table 3.1 shows our official results in terms of mean accuracy, together with the highest and the lowest performers ("MAX" and "MIN").

It can be observed that our runs are not very successful. Below, we report on an initial failure analysis.

Unfortunately, our systems returned the same nonrelevant tweets for many of our test post tweets. Figures 4.1 and 4.2 show the top three tweets that were returned most frequently in runs SLSTC-J-R2 and SLSTC-J-R3, respectively. We investigated why these tweets were returned regardless of what the test tweet was.

Table 3.1: Official results (mean accuracy)

	Acc_{L2}@1	Acc_{L2}@5	$Acc_{L1,L2}$@1	$Acc_{L1,L2}$@5
SLSTC-J-R1	0.0381	0.0364	0.1644	0.1650
SLSTC-J-R2	0.0782	0.0332	0.3416	0.1795
SLSTC-J-R3	0.0054	0.0032	0.0391	0.0196
MAX	0.4574	0.3583	0.7817	0.7050
MIN	0.0054	0.0032	0.0391	0.0196

Figure 4.1: Top 3 tweets appearing in the results of run 2

Table 4.1 and Table 4.2 show examples of test post tweets, the nouns extracted from them, the top three highest scoring words within the subgraphs, together with the PageRank, idf and the product of the two for each of the subgraph word. The accuracy of both results are relatively low: 0.04 for the former one, 0.16 for the latter one in $Acc_{L1,L2}$@5. It can be observed that the most of the enabled words are topically unrelated to the tweet, except for "バスターミナル" (bus terminal) which is somewhat related to the nouns "各駅停車" (local train) and "神戸三宮" (Kobe Sannomiya Station). Moreover, the word "PC ゲーム" (PC game) was enabled for both tweets. In this way, it appears that similar words were enabled regardless of the test tweet, probably because we expanded our nodes too aggressively using Eqs. (2.6) and (2.7). That is, our subgraphs for the different test topics were all similar. In addition, it can be observed that the accuracy of noun extraction based on MeCab is low.

4. CONCLUSIONS AND FUTURE WORK

We participated in the Japanese subtask of the NTCIR-12 STC task using approaches based on neural networks and word co-occurrence networks. Unfortunately, our results were not satisfactory. Our initial failure analysis shows that for the word co-occurrence networks, morphological analysis errors and the inclusion of too many words in the subgraphs were the main bottlenecks. As future work, we would like to introduce a weighting scheme for the edges of the subgraphs to improve the accuracy, and to customise the morphological analysis dictionary.

Table 4.1: top 3 PageRank * IDF score words in subnetwork for the test post tweet ID:566650533423763456

test post tweet	nouns extracted	words in subgraph with highest scores	PageRank	IDF	PageRank*IDF
ゆうくりっどさんが言いたいことにプラスして言及してくれてた	さんが, プラス, して	PC ゲーム	0.0000296	4.75	0.000141
		トレシュー	0.0000240	5.23	0.000125
		女川	0.0000237	5.23	0.000124

Table 4.2: top 3 PageRank * IDF score words in subnetwork for the test post tweet ID:580602764306333696

test post tweet	nouns extracted	words in subgraph with highest scores	PageRank	IDF	PageRank*IDF
【自動】 お待たせしました。 7 号線、各駅停車、神戸三宮行き ただいま発車します。	自動, 7 号, 各 駅 停 車, 神戸三宮	ゃん いってらっしゃいだよっ	0.0000208	5.23	0.000109
		バスターミナル	0.0000187	5.23	0.0000975
		PC ゲーム	0.0000187	4.75	0.0000887

5. REFERENCES

[1] Shang, L., Sakai, T., Lu, Z., Li, H., Higashinaka, R. and Miyao, U., Overview of the NTCIR-12 Short Text Conversation Task, Proceedings of NTCIR-12, 2016.

[2] Google Word2Vec, https://code.google.com/p/word2vec/

[3] MeCab, http://taku910.github.io/mecab/

[4] Bahman Kermanshahi, Construction and Application of Neural Network, Shokodo

[5] Sergey, B., Rajeev, M., and Terry, W., What can you do with a web in your pocket, Data Engineering Bulletin, Vol. 21, pp. 37-47, 1998.

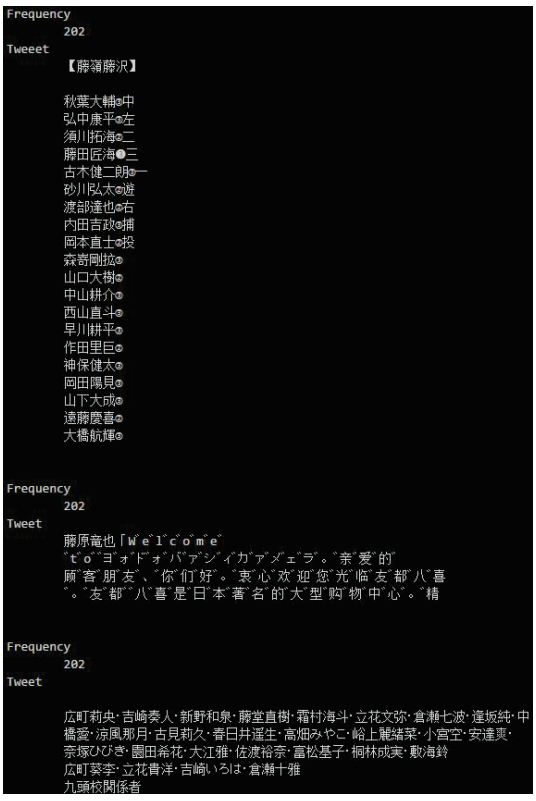

Figure 4.2: Top 3 tweets appearing in the results of run 3