Monitoring Geographical Entities with Temporal Awareness in Tweets

Koji Matsuda[1], Mizuki Sango[2], Naoaki Okazaki[1], and Kentaro Inui[1]

[1] Tohoku University
matsuda@ecei.tohoku.ac.jp
[2] Tokyo Institute of Technology

Abstract. To extract real-time information referring to a specific place from social network service texts such as tweets, it is necessary to analyze the temporal semantics of the reference. To solve this problem, we created a corpus with multiple annotations for more than 10,000 tweets using crowdsourcing. We constructed an automatic analysis model based on multiple neural networks and compared their characteristics. Our dataset and codes are released in our website [3].

1 Introduction

People often mention geographical entities (e.g., cities, tourist spots and public facilities) on social networking services (SNSs) expressing their present or past experiences and future plans for visits. Let us consider the following example.

(1) I went to Sendai yesterday, but I'm going to Nagoya today.

This example has references to two places, Sendai and Nagoya. However, the time recognition of the author for each place is different: Sendai is a *past* location, whereas Nagoya is a *present* location.

In applications such as text mining and marketing for tourism, it is crucial to distinguish such temporal references. By resolving temporal aspects, we can extract the opinions of people who actually stay(ed) at a certain location. In addition, recognizing individuals who are willing to go to a certain location facilitates targeted advertisements for potential visitors.

In this paper, we discuss tasks to monitor references to geographical entities that appear in the texts of SNSs. In particular, we work on the task of detecting users who are present in a location at the moment and detecting references including the intention to go a location. To realize this goal, we need to address at least two problems. First, we need to disambiguate references in the text into geographical entities because a reference can refer to multiple locations or to non-geographical entities. Second, we have another type of ambiguity problem concerning the time recognition of the author, i.e., whether and when the author (will) stay(ed) at the location. This paper discusses the latter problem, which recognizes the temporal relation between a geographical entity and an author given in a tweet.

[3] http://www.cl.ecei.tohoku.ac.jp/~matsuda/TA_corpus/

Our task is closely related to *Temporal Awareness* [7] (*TA*, hereafter), where location references and temporal polarities in tweets are identified. However, to avoid the ambiguity effect, we assume that the detection of the location mentioned has already been completed and create an annotated corpus focusing on the classification of the temporal relation. To emulate this situation, we target tweets that contain predefined nouns that are known to be location references. Even if this is done for a limited number of targets, it is interesting to see whether linguistic features are learnable.

In addition, we created a model that automatically analyzes *TA* using this corpus. We use a model motivated by target-dependent sentiment classification [3], which is a variation of short text classification that incorporates target information.

The contributions of this paper are two folds.

- We designed an annotation scheme focusing on *TA*. We annotated more than 10,000 tweets using a crowdsourcing platform. The quality of the annotation was confirmed to be high, which indicates clearly that the task was properly designed.
- We built a model using a state-of-the-art method based on neural networks. We show its quantitative performance. In addition, we conducted experiments with cross domains to demonstrate the performance of the *TA* for unknown targets.

2 Related Work

Li et al. [7] proposed an end-to-end model to extract expressions representing places with time labels using the framework of sequence labeling.

They proposed a model that uses sequence labeling to simultaneously extract the references representing places and the time labels. The categories of location references they deal with are diverse and cover various expressions such as facility names and place names referring to unique entities. However, their model does not focus on temporal relationships, because it solves multiple tasks, such as ambiguities of references referring to places and temporal relationships.

In addition, we assume that as a practical usage scenario we should gather tweets mentioning specific places. However, it is not easy to gather tweets that refer to all entities with an open vocabulary. For example, because Twitter's streaming API can only obtain a very small sample of tweets, it is not appropriate to monitor references to specific entities with high coverage. In contrast, a search API is more realistic for entity monitoring because it provides data with a high coverage. In addition, the data created by Li et al. have not been verified by multiple annotations. Therefore, the validity of the design of their annotation scheme cannot be evaluated.

Recently, Huang et al. created a corpus annotating event information to summarize news and generate timelines [2]. In their corpus, the temporal status of an event is annotated to the a major social event (in particular, a civil unrest event) described in the news text whether the event has actually occurred or is going to occur. Their goal is similar to ours; however, we differ in that we aim to estimate the intentions of people's daily lives rather than to detect major social events. In addition, they target news text the text which is different in nature from our user generated text.

Our work is linked to the TIMEBANK [8] and TempEval [12] efforts; however, we consider lightweight corpus specifications. To scale the annotation, we created a simple annotation guideline and user interface and proposed a framework that allows annotators who are not experts to do high-quality work.

Readers may find this task is similar to Factuality analysis [9] in the task is to predict whether the events mentioned in the sentence correspond to actual events that have occurred. Typical Factuality analysis is intended for events represented explicitly in the text. However, as you can see in the example below, the fact that someone (will) stay(ed) in a certain place is not always explicitly written in the text.

(2) I lost my way in Sendai station.

In this sentence, the interpretation that the author visited Sendai station is reasonable; however, because it is implicitly written, it cannot be handled in the existing framework. To capture such an implicit event, we focus on the location reference rather than on the events explicitly mentioned in the text.

This task can also be seen as a short text classification problem [4, 5, 10]. However, it is reasonable to view our problem as a target-dependent short-document classification problem. This is because it is possible to assume that multiple targets appear in one short text, as in Example (1). In particular, our task is close to target-dependent sentiment analysis. In the target-dependent short-text classification problem, it has been reported that neural models using convolutional neural networks (CNNs) and recurrent neural networks (RNNs) show high performance [3, 1, 13]. Therefore, this study explores neural models in Section 4.

3 Dataset Construction

In this task, how to create data is also a big issue. First, an expression that points to a specific place is not limited to a proper noun. If the ambiguity can be resolved in some way, some general nouns, such as "hospital" or "city-hall", can also be considered as monitoring targets.

We focus on a realistic situation in which the entities to be monitored are known. Given this situation, we create data focusing on several pre-selected location. In general, it is necessary to create annotated data for each entity; however, the general linguistic meaning independent of entity can be learned from data for other entities. This suggests that it is possible to identify the TA with some degree of performance for both learned and unknown targets from annotated data for fixed targets. Of course, by annotating the target you want to monitor, it is possible to improve its performance. To experimentally verify this prediction, we adopted a policy to use closed vocabulary and to annotate a large number of instances for each target.

When annotating the temporal nature concurrently with picking up identifying the target reference, the attention of the annotator becomes distracted and high-quality annotation is not realistic. Therefore, we create data with closed vocabulary for nouns that are likely to be references to places. As a usage scenario, there may be situations where the target to be monitored is known and a certain amount of training data on the target can be obtained. Therefore, a closed setting is also meaningful.

Specifically, we first compiled a list of Japanese nouns representing a place (a location) and annotated only Japanese tweets containing at least one noun (one location

Table 1. Label set and description of annotating *TA*.

Label	Description	Example (Target is **bold**)
`Present`	The author is at or near the location represented by the target.	The **Eiffel Tower** is beautiful.
`Past`	The author was in the place represented by the target.	Going to the **Eiffel Tower** gave me memories for the weekend.
`Future`	The author is not currently in the place represented by the target word, but seems to be going there now.	I am going to the **Eiffel Tower** from now.
`Non-Temporal`	The author has never been in the place represented by the target word and has no plans to go.	What time does the **Eiffel Tower** open?
`Non-Mention`	Author does not mention places represented by the target word.	I watched the movie **"Eiffel Tower"**.

references). This list contains five proper nouns and five common nouns that we consider equally, including the location name, facility name, and tourist spot name, and that are chosen based on the criterion that an interpretation other than a reference to a place (a personal name or organization name) rarely occurs [4].

For 1200 Japanese tweets for each target, seven workers annotated based on the label set in Table 1, which is based on Li et al. [7]. To minimize the annotator's load as much as possible when collecting annotations based on crowdsourcing, we tried to make the specifications in the annotation as simple as possible. In the actual annotation, we presented the tweet text, and the target noun and asked the annotator to choose one of the choices for the tweet author's time recognition from the candidate labels for the entity represented by that noun, as showed in Fig. 1. To eliminate malicious annotators, we did not collect user annotations that could not be answered correctly by mixing check questions with correct answers at a rate of 1 out of every 18 tweets. We paid approximately 0.4 JPY (approximately 0.34 US cents) to the annotators for annotating one tweet. We used the Yahoo! Crowdsourcing platform to collect the annotations. The basic statistical information of the created corpus is shown in Table 2.

To investigate the quality of the annotation, we calculated the number of annotates are consistent for each annotated tweet. This result is shown in Table 2. As a result, we found that 93% of the tweets coincided with the labels of five or more people out of seven. This indicates that a relatively stable annotation can be performed. In addition, there was no major bias in the distribution of the annotated labels. This result suggests that our label set is stable.

[4] We used the following 10 nouns as targets: "Akihabara", "Kiyomizu-dera (Kiyomizu Temple)", "Shibuya-eki (Shibuya station)", "Sky Tree", "Sendai", "shiyakusyo (city hall)", "kousaten (crossing)", "byouin (hospital)", "kaisatsu (ticket gate)" and "doubtsuen (zoo)".

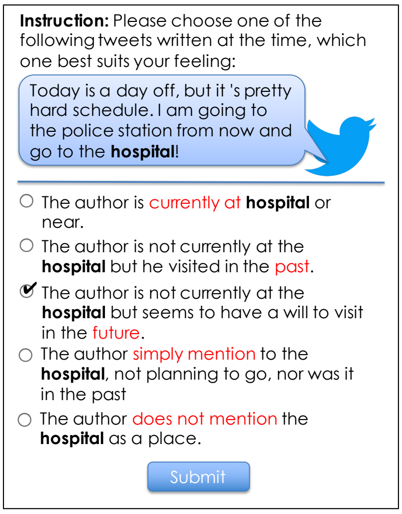

Fig. 1. Example of the annotation user interface in English. (The actual work was performed in Japanese.)

Fig. 2. Dataset Statistics.

Total tweets		12318
Agreement	7 votes	2212(18%)
	6 votes	5452(44%)
	5 votes	3797(31%)
Label	`Present`	2413(20%)
	`Past`	2342(19%)
	`Future`	2134(17%)
	`Non-Temporal`	4416(36%)
	`Non-Mention`	962(8%)

Agreement Analysis Table 2 shows the distribution of the agreement aggregated for each target. For most of the targets, it can be seen that five annotations match for more than 90% of the tweets. However, the agreement for "ticket gate" is lower than that of the other targets. Because there are relatively few people who make long-term visits to ticket gates, most tweets reported that they had simply passed the ticket gate.

(3) I passed by a ticket gate with a person similar to Jeson.

In Example (3), three workers annotated the reference as `Present`, but four workers annotated it as `Past`. Both interpretations are reasonable; however, we set the threshold to five votes and interpretations with values less than that ware not used in the automatic analysis experiment.

Label Distribution The distribution of labels in the data set is shown in Table 3. It was found that the distribution of labels differed for each target.

In particular, instances that refer to sightseeing spots (e.g. "Kiyomizu Temple", "Tokyo Skytree" and "zoo") were often labeled `Non-Temporal`. From observations of several instances, we see that there were many reference to sightseeing spots seen on television that ware just impressions. In addition, there were a relatively large number of instances where "Sendai" and "zoo" were labeled as `Non-Mention`; however, this was largely influenced by compound nouns, metaphorical expressions[5] and ambiguity of the semantic class (e.g., the organization or location).

[5] In Japanese, "zoo" is also used as a metaphor to indicate a lively appearance.

Table 2. The annotation agreement rate calculated for each target. Numbers in parentheses indicate percentages. The value of the last column indicates the percentage of instances where an agreement of five votes or more was obtained.

			Agreement			
Target (in English)	proper?	# of tweets	7 votes	6 votes	5 votes	$\geq$ 5 votes
"Akihabara"	✓	1254	315 (0.25)	529 (0.42)	345 (0.28)	0.948
"city-hall"		1204	235 (0.20)	522 (0.43)	367 (0.30)	0.934
"crossing"		1233	235 (0.20)	522 (0.43)	367 (0.30)	0.926
"hospital"		1257	325 (0.26)	532 (0.42)	318 (0.25)	0.934
"Kiyomizu Temple"	✓	1199	233 (0.19)	566 (0.47)	339 (0.28)	0.949
"Sendai"	✓	1240	201 (0.16)	577 (0.47)	383 (0.31)	0.936
"Shibuya Station"	✓	1214	219 (0.18)	538 (0.44)	383 (0.32)	0.939
"Tokyo Skytree"	✓	1220	197 (0.16)	553 (0.45)	373 (0.31)	0.904
"ticket gate"		1257	150 (0.12)	496 (0.39)	469 (0.37)	0.887
"zoo"		1240	181 (0.15)	610 (0.49)	363 (0.29)	0.930
Total		12318	2212 (0.18)	5452 (0.44)	3797 (0.31)	0.930

(4) The representative of the Miyagi Prefecture is decided by the Sendai Ikuei High School.

In Example (4), a part of the high school name mentioned in the sentence contains the place name Sendai; however, it does not represent Sendai itself. Because it is difficult to automatically exclude such instances, we excluded instances with the `Non-Mention` label in the automatic analysis experiment in this study.

From this data, we removed the instances that were labeled `Non-mention` and divided the dataset into 700 training data, 100 development data and 100 test data for each target. We used this split of the data to train the model and to validate its performance.

4 Automatic Analysis of *Temporal Awareness*

To analyze the *TA* automatically, we formulated the problem as a target-dependent text classification as follows. In our models, we calculate the left and right context representation $v_r \in \mathbb{R}^M, v_l \in \mathbb{R}^M$ separately. These vectors are calculated via an "Encoder" module such as CNN or BiLSTM (bidirectional long short-term memory) from a sequence of word embedding vectors. Then, the concatenation of these representations is used to calculate the label distribution $y \in \{\texttt{Present}, \texttt{Past}, \texttt{Future}, \texttt{Non-Temporal}\}$ based on a feed-forward neural network.

4.1 Incorporation of Target Information via the Network Structure

In this task, it was expected that a chain of words and expressions that appear in the target's neighborhood context would be a large clue. Therefore, giving clues by the position of the target is natural. A large portion of the example was expected to determined

Table 3. Detailed distribution of the final labels of our dataset. The numbers in parentheses indicate the percentages.

Target (in English)	# of tweets	Labels				
		`Present`	`Past`	`Future`	`Non-Temporal`	`Non-Mention`
"Akihabara"	1254	281 (0.22)	182 (0.15)	405 (0.32)	370 (0.30)	13 (0.01)
"city-hall"	1204	159 (0.13)	169 (0.14)	291 (0.24)	495 (0.41)	87 (0.07)
"crossing"	1233	303 (0.25)	403 (0.33)	28 (0.02)	413 (0.33)	83 (0.07)
"hospital"	1257	202 (0.16)	269 (0.21)	417 (0.33)	323 (0.26)	39 (0.03)
"Kiyomizu Temple"	1199	120 (0.10)	275 (0.23)	219 (0.18)	555 (0.46)	29 (0.02)
"Sendai"	1240	177 (0.14)	108 (0.09)	231 (0.19)	440 (0.35)	276 (0.22)
"Shibuya Station"	1214	451 (0.37)	277 (0.23)	90 (0.07)	389 (0.32)	1 (0.00)
"Tokyo Skytree"	1220	212 (0.17)	125 (0.10)	215 (0.18)	532 (0.44)	130 (0.11)
"ticket gate"	1257	438 (0.35)	425 (0.34)	50 (0.04)	305 (0.24)	28 (0.02)
"zoo"	1240	70 (0.06)	109 (0.09)	188 (0.15)	594 (0.48)	276 (0.22)
Total	12318	2413 (0.20)	2342 (0.19)	2134 (0.17)	4416 (0.36)	962 (0.08)

the from left or right context of the target. To incorporate the target information into this model, we separately computed the vector representation of the target-dependent sequence splitting. We considered the following two architectures in our experiment.

Flat This model encodes a full of sentence at once without considering the target in context.

Target-Dependent We also tried to introduce target information following Tang et al. [11] which is a state-of-the-art model of target-dependent sentiment classification. In this model, the left context and right context of the target are encoded separately and concatenated as $v = [v_l; v_r]$.

4.2 Encoders

We used the following Encoders to compare the classification performance.

Averaging Encoder The Averaging Encoder computes the averages of the words in a sentence. We used this model as baseline-encoding model. This model does not consider word ordering or collocation but has good performance in many tasks and can be a good baseline.

Convolutional Encoder We also used the CNN encoder based on Kim [5]. In this model, we obtain the vector representation of the sentence v via fixed size convolution and max-pooling.

Let $\mathbf{x} \in \mathbb{R}^k$ be a k-dimensional word vector. A sentence can be expressed as follows using a concatenation of word vectors:

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus ... \oplus \mathbf{x}_n \tag{1}$$

In this equation, $\oplus$ is the concatenation operator of the vector. Let $\mathbf{x}_{i:i+j}$ be the concatenation of the i th word to the $i + j$ th word in the sentence. Here we introduce the filter matrix $\mathbf{W} \in \mathbb{R}^{hk \times L}$. H represents the size of the filter (which corresponds to n in the n-gram model); in this paper, we used two words. L represents the number of filters to be applied. The result of applying the l th filter (the vector of the l th row of $\mathbf{W}$) to the i th word is calculated as follows:

$$c_{i,l} = f(\mathbf{W}_{\cdot l} \cdot \mathbf{x}_{i:i+h-1} + b) \tag{2}$$

Where $b \in \mathbb{R}$ is the bias term. f is a nonlinear function; we used the sigmoid function in this paper. This filter is applied to all positions $(1, 2, ...n - h + 1)$ in the sentence and the following feature map $\mathbf{c} \in \mathbb{R}^{n-h+1}$:

$$\mathbf{c}_l = [c_{1,l}, c_{2,l}, ..., c_{n-h+1,l}] \tag{3}$$

Then, the maximum value of the feature map vector is extracted via max-over-time pooling.

$$\hat{c}_l = max(\mathbf{c}_l) \tag{4}$$

This procedure is performed for each filter vector and is used as a representation of the sentence.

Bi-Directional LSTM (BiLSTM) Encoder For the BiLSTM encoder, where u_i is a N dimensional input embedding of a word, h_{i-1} is the previous output, and s_{i-1} is the previous cell state.

$$h_i, s_i = lstm(u_i, h_{i-1}, s_{i-1}) \tag{5}$$

For given text, the LSTM encoder is applied recursively to sequence from left-to-right. Our model adopted the Bi-directional LSTM model, which is concatenate of the final output state of left-to-right encoding $\overrightarrow{h}$ and right-to-left encoding $\overleftarrow{h}$ as $v_{bilstm} = [\overrightarrow{h}; \overleftarrow{h}]$. We also tried another model that incorporated the attention mechanism; however, the performance did not improve.

5 Experiment

We experimentally examined three types of options on the sentence encoding method and two network architectures for introducing the target information. As a baseline, we trained logistic the regression model with unigram and bigram features. We experimented with the following three settings.

In-domain In this setting, training data for all target words including the target word used for the test are used for training. This setting assumes that labeled data of target word that want to monitor can be obtained.

Table 4. Overall result (accuracy) of different encoders and their composition.

Architecture	Encoder	In-domain	10%	Cross-domain
Majority Baseline		0.390	0.390	0.390
MaxEnt Model(Uni+Bi)		0.673	0.639	0.593
Flat	Averaging	0.606	0.554	0.554
	BiLSTM	0.599	0.490	0.516
	CNN	0.607	0.586	0.553
Target Dependent	Averaging	0.583	0.564	0.548
	BiLSTM	0.684	0.634	0.609
	CNN	0.669	0.628	0.591

Cross-domain In this setting, training data of the target word used for the test are not used. This is an experiment assuming the case where the target entity is unknown.

10% samples In addition, we considered an intermediate situation assuming that a small amount of training data could be prepared for the target. Specifically, 10% of the data (70 instances) was sampled from the training data for the target and used in addition to the training data of other targets.

In all settings, the classification accuracy was used as the evaluation metric.

5.1 Detailed Setting

For all models, we used 300 dimensional `word2vec` embedding, learned from the Japanese twitter corpus, as the initial value for embedding. For optimization, ADAM [6] was used with default hyper-parameters, the size of a mini batch was fixed to 100, and the dropout rate of each layer was fixed to 0.5. To optimize each model equally, we adopted "early stopping" technique to train all models. When the accuracy for the development set was not updated for more than 10 epochs, we assigned a label to the test data using the maximum accuracy model on the development set.

5.2 Result

Table 4 shows the classification performance of our models. Flat architectures were found to be ineffective. In addition, the encoder that separately encodes the right and left sides of the target shows higher performance than the flatly encoding model. Interestingly, we observed that the encoders based on BiLSTM and CNN have very different architectures, but achieved similar classification performances. In addition, even in the setting of the cross-domain, our models achieved a certain level of performance, which suggests that knowledge transfer between targets is possible. Finally, by combining Target Dependent encoding and BiLSTM encoder, we found that it is competitive or slightly better performance than the baseline maximum entropy model.

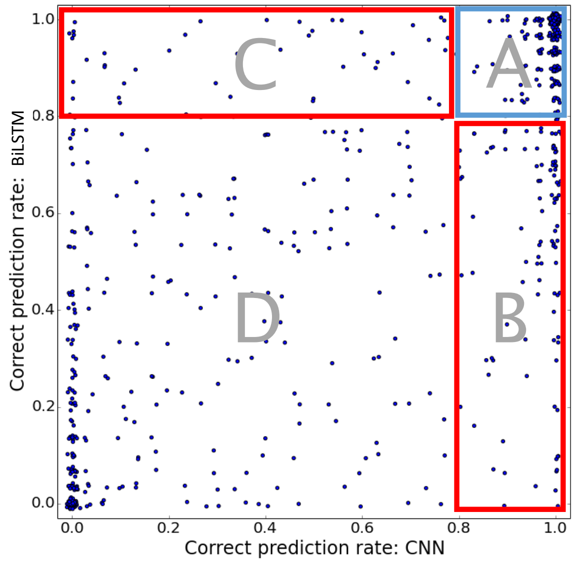

Fig. 3. Scatter plot of individual instances in development data, correct prediction rate of BiLSTM and CNN encoders over 30 different initialization.

6 Analyzing Characteristics of Different Encoders with an Initialization Test

We hypothesize that the encoders based on BiLSTM and CNN have different characteristics for the context utilization.

To compare different models, initial values in the learning stage of each model were varied randomly and each instance was analyzed to see whether it was accidentally correct or stably correct. First, we defined a metric called the correct prediction rate (CPR). This is a value defined for each instance that indicates whether the model learned from a number of initial values among the changed initial values can be corrected. We calculated the CPR for each instance for the CNN and BiLSTM models.

More specifically, the following method was used. We trained CNN-based target-dependent model and BiLSTM-based target-dependent model with 30 different initializations; we plot each example of development data in Fig.3. The horizontal axis is the CPR of the CNN-based encoder, and the vertical axis is the CPR of the BiLSTM-based encoder. In this figure, the highlighted region in upper right means correctly predicted in most initializations, regardless of the initialization of the model. We found that a large portion of the examples are closer to both sides on the vertical axis, but not on the horizontal axis, this means that the CNN-based encoder is relatively more robust to initialization than the BiLSTM encoder.

We divided this scattergram into four regions, and investigated to which area each instances of development data belonged. We found that (A) 42.8% of examples (462/958) had over 80% CPR in both encoders; (B) 12.1% of examples (116/958) had over 80%

CPR in the CNN encoder, but less than 80% CPR in the BiLSTM encoder; (C) 5.5% of examples (53/958) had over 80% CPR in the BiLSTM encoder, but less than 80% CPR in the CNN encoder; and (D) 36.8% of examples (353/958) had less than 80% CPR in both encoders.

To examine which examples are differently encoded, we employ the linguistic annotator to annotate with *TA* label and their target reference, in addition, we consulted to give label to a clue words that need to predict *TA* label. This annotation was performed on randomly shuffled (B) + (C) portion of development data, and actual *TA* label ware hide from the annotator. As a result, clues were annotated in 58% of instances included in area (B), whereas in cases included in area (C) clues were only found in 32% of instances annotated. This result suggests that CNN can successfully encode sentences with clearer clues, whereas BiLSTM is better suited to handle ambiguous clues such as chain of function words.

7 Conclusions

In this paper, we addressed the task of analyzing the Temporal Awareness for location references. As a result of crowdsourcing annotation, the agreement between annotations was high, indicating that the task was properly designed. In addition, we constructed encoders based on BiLSTM and CNN and compared them. It became clear that BiLSTM can handle blurred clues such as linkage of function words better. Investigating of a model integrated with location name extraction and disambiguation is left as a future task.

Acknowledgments

This work was partially supported by *Research and Development on Real World Big Data Integration and Analysis*, MEXT, Japan.

References

1. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the ACL (2014)
2. Huang, R., Cases, I., Jurafsky, D., Condoravdi, C., Riloff, E.: Distinguishing past, on-going, and future events: The eventstatus corpus. In: EMNLP (2016)
3. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: Proceedings of the ACL-HLT (2011)
4. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058 (2014)
5. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ICLR (2014), `http://arxiv.org/abs/1412.6980`
7. Li, C., Sun, A.: Fine-grained location extraction from tweets with temporal awareness. In: Proceedings of the SIGIR (2014)

8. Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al.: The timebank corpus. In: Corpus linguistics. vol. 2003, p. 40 (2003)
9. Saurí, R., Pustejovsky, J.: Are you sure that this happened? assessing the factuality degree of events in text. Computational Linguistics 38(2), 261–299 (Jun 2012), http://dx.doi.org/10.1162/COLI_a_00096
10. Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the SIGIR (2015)
11. Tang, D., Qin, B., Feng, X., Liu, T.: Effective lstms for target-dependent sentiment classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 3298–3307. The COLING 2016 Organizing Committee, Osaka, Japan (December 2016), http://aclweb.org/anthology/C16-1311
12. Verhagen, M., Sauri, R., Caselli, T., Pustejovsky, J.: Semeval-2010 task 13: Tempeval-2. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 57–62. Association for Computational Linguistics (2010)
13. Vo, D.T., Zhang, Y.: Target-dependent twitter sentiment classification with rich automatic features. In: Proceedings of the IJCAI (2015)