TUA1 at the NTCIR-13 Actionable Knowledge Graph Task: Sampling Related Actions from Online Searching

Xin Kang
kang-xin@is.tokushima-u.ac.jp

Yunong Wu
wuyunong@tokushima-u.ac.jp

Fuji Ren
ren@is.tokushima-u.ac.jp

Faculty of Engineering, Tokushima University
2-1 Minamijyousanjima-cho, Tokushima 770-8506, Japan

ABSTRACT

This paper details our partition in the Action Mining (AM) subtask of NTCIR-13 Actionable Knowledge Graph (AKG) Task. Our work focuses on sequentially sampling the most related actions for any named entity based on online search results. We propose three criteria, i.e. significance, representativeness, and diverseness, for evaluating the relatedness of candidate actions in the search results. We analyze the quality of sampled actions from different online search strategies. The experiment results suggest that our method is effective for generating a sequence of related actions for named entities.

Team Name

TUA1

Subtasks

Action Mining (AM) Subtask

Keywords

Entity Related Action, Action Mining, Knowledge Graph

1. INTRODUCTION

In the Action Mining (AM) subtask of the NTCIR-13 Actionable Knowledge Graph (AKG) Task [2], the TUA1 team focuses on a sampling method which sequentially selects the most related actions for an entity from online search results. Our sampling approach is borrowed from the sampling strategies [8] in the active learning algorithm. For example, given the entity *adoption*, our method generates highly-related actions like "*adopt* children of a different race and/or culture" which demonstrates a typical action of *adoption*, and "*see* Santosky v. Kramer, 455 U.S. 745 (1982)" which leads to the critical procedures related to *adoption*. The sampling strategy is featured by three criteria for evaluating the probability of being selected as a related action, i.e. the significance of an action, the representativeness of an action, and the diverseness of an action from the other selected actions.

A general action consists two parts, i.e. a verb characterizing an activity and a description about the objective. For example, *adopt* and *see* are the activities and "children of a different race" and "Santosky v. Kramer" are the objectives. To generate an action, our method firstly selects the most related verb from online search results, and then selects the most related objective.

All candidate actions for an entity are extracted from the retrieved documents through an online search, with the entity as the search query. The retrieved documents constitute a corpus D. For each sentence in the corpus, we perform syntactic parsing to extract a candidate verb and the corresponding objective description.

With a set of candidate verbs in U, we sequentially draw a verb sample v from U and put it into the target verb list V. The sampling strategy is based on three criteria, i.e. significance, representativeness, and diverseness. A verb $v \in U$ is considered to be significant if it has been observed for many times through the corpus D but should not be a common verb like "be", "do" which does not contain much specific meaning. A verb $v \in U$ is considered to be representative if there are many verbs in D which express close semantic meanings to v. A verb $v \in U$ is considered to be diverse with respect to selected verbs in V if its semantically most close verb $v' \in V$ still has a distinct difference to v.

For each selected verb $v \in V$, we construct a candidate objective set N and sequentially draw an objective o from N then move it into the target objective set O. The sampling strategy is similar to verb selection, except that we do not consider the significance of candidate objectives. This is because that evaluating the significance of an objective description requires learning a separate regression model, which is out of the scope of this work. We consider all candidate objectives with the same significance and select the objective $o \in N$ by evaluating its representativeness in D and diverseness from O.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work of action mining. Section 3 describes our sampling method for generating related actions in detail. We report our experiment and analyze results in section 4, and conclude this paper in section 5.

2. RELATED WORK

A very close study to the Action Mining (AM) subtask is the intent mining in search queries [1, 3, 5, 7, 9, 10, 11]. Its motivation is similar to AM as to understand the users' latent intents in search queries. The distinction is also obvious, although search queries are short, they still contain rich information, such as proper nouns, adjectives, verbs, while an entity in AM is specified only with its name and category. More importantly, there are huge number of query logs every day stored by search engine companies, which provide the basis for mining users' intent behaviours. For AM, the current available language resource is limited and requires further manual explorations. Last but not the least, the intent class for query intent mining is usually predefined, which is suitable for search engines to refine their retrieved results. However, the action class in AM is open. Therefore, AM is an information retrieval problem while query intent mining is usually a classification problem.

3. ACTION SAMPLING METHODS

The problem of AM can be formally defined as given an entity e, explore any external resource D to generate a ranked list of actions (V, O), in which V and O are verbs and objective descriptions in these actions.

In the first step of related action sampling, we perform an online search with entity e as the query string to retrieve the related documents in a corpus D. Depending on the search strategy, we have constructed three distinct corpora $D^{(Q)}$, $D^{(U)}$, and $D^{(G)}$, each with a different set of documents. Corpus $D^{(Q)}$ is constructed by querying the Twitter Search API to retrieve at most 1,500 pages of entity-related Tweets. Corpus $D^{(U)}$ is constructed by first querying the Twitter User Search API to collect at most 1,000 entity-related users, then querying the Twitter User Timeline API to retrieve these users' latest Tweets of at most 16 pages. Corpus $D^{(G)}$ is constructed by first querying Google to retrieve 3 pages of entity-related search results, then downloading and extracting the documents from these search results. All retrieved documents in other language rather than English are filtered out. The examples of retrieved documents are shown in Table 1.

In the second step, we extract all verbs from D to construct a verb set U, and sequentially draw the related verb samples v from U to construct a ranked verb list V[1]. The sampling strategy for v is based on three criteria, i.e. significance, representativeness, and diverseness, of v with respect to all verbs in D and all selected verbs in V. The calculations of three criteria are illustrated as follows.

The **significance of verb** v evaluates the degree of popularity of v in D. Its calculation consists of two parts as

$$\mathrm{s}(v) = \mathrm{tf}(v) \times \mathrm{idf}(v), \tag{1}$$

in which $\mathrm{tf}(v)$ is the term frequency of v among all verbs v' in U as

$$\mathrm{tf}(v) = \frac{f_v}{\sum_{v' \in U} f_{v'}} \tag{2}$$

and $\mathrm{idf}(v)$ is the inverse document frequency of v in D as

$$\mathrm{idf}(v) = \log \frac{|D|}{|\{d | v \in d, d \in D\}|}. \tag{3}$$

Different from the traditional definition of term frequency, $\mathrm{tf}(v)$ is large for v if it is observed for many times in D. The inverse document frequency $\mathrm{idf}(v)$ is large for v if it is observed in very few documents in D. For entity e, an important verb v should describe a popular action among the retrieved related documents D. Therefore, we need the term frequency $\mathrm{tf}(v)$ to be as large as possible. However, there are some very common verbs, such as "be", "do", which render very large term frequencies but do not suggest any meaningful action. To filter them out, we employ the inverse document frequency which renders very small values for these very common verbs. In consequence, we need the inverse document frequency $\mathrm{idf}(v)$ to be also as large as possible. We combine these together to calculate the significance criterion $\mathrm{s}(v)$ for a verb v.

The **representativeness of verb** v evaluates the degree of averaged similarity of v to all verbs in U. The similarity between two verbs v and v' refers to the cosine similarity of the vector representation of semantic meanings of these two verbs. Its calculation is as follows

$$\mathrm{sim}(v, v') = \frac{\vec{v} \cdot \vec{v'}}{||\vec{v}|| \times ||\vec{v'}||}, \tag{4}$$

in which $\vec{v}$ and $\vec{v'}$ are the vector representation of semantic meanings of v and v', respectively. The vector representations of verb semantic meanings are generated by the Enwiki Word2vec model with 1000 dimensions[4]. The representativeness of a verb v is therefore calculated as

$$\mathrm{r}(v) = \frac{1}{|U|} \sum_{v' \in U} \mathrm{sim}(v, v'), \tag{5}$$

which is the averaged similarity of v to all verbs in U. If there are many verbs v' in the retrieved related documents D which share similar semantic meaning with verb v, we can conclude that this verb represents the action of the other similar verbs v'. Therefore, selecting verb v with a large representativeness criterion r(v) is equivalent to sampling a representative action for entity e.

The **diverseness of verb** v evaluates the degree of dissimilarity of v to all verbs in V, where V is the list of already selected verbs. For a candidate verb $v \in U$ and a selected verb $v' \in V$, we employ the minus of similarity in Eq. 4 to represent their dis-similarity

$$\overline{\mathrm{sim}}(v, v') = -\mathrm{sim}(v, v'). \tag{6}$$

After selecting a representative verb $v' \in V$, we want to avoid sampling another verb v with very similar semantic meaning to it. In this sense, we calculate the diverseness of v as

$$\mathrm{d}(v) = \min_{v' \in V} \overline{\mathrm{sim}}(v, v'). \tag{7}$$

The $\overline{\mathrm{sim}}(v, v')$ evaluates degree of semantic difference between verbs v and v'. A large $\overline{\mathrm{sim}}(v, v')$ indicates a huge semantic difference between v and v'. By taking the minimum $\overline{\mathrm{sim}}(v, v')$ in Eq. 7, we can evaluate the semantic difference between a candidate verb v and its semantically most similar verb in the selected verb list V. If criterion $\mathrm{d}(v)$ is large, we can conclude that the semantic meaning of verb v is very different from the semantic meanings of all selected verbs $v' \in V$, and therefore is a diverse enough as a candidate for selection. This technique has also been employed for learning the salient samples for SNS message polarity classification [6].

With these three criteria, we are able to sample related verbs from U to construct a ranked verb list V. The sampling algorithm is depicted in Algorithm 1. At the beginning, we initialize U to be the set of all verbs in the retrieved corpus D, and initialize V to be an empty list Φ. To select n related verbs, we repeat the sampling procedure for n times. In each loop, we calculate the three criteria for every verb $v \in U$, and sum them up to generate the sampling criterion. As illustrated above, a related verb should have large values in all three criteria, the algorithm selects verb v with the largest sum of three criteria at the end of this loop. The

[1]Because this process is general for $D^{(Q)}$, $D^{(U)}$, and $D^{(G)}$, we do not specify the upper scripts for D, U, V, and v here.

Entity	Strategy	Document
Adoption	Q	RT @RopeAndAnchorLS: Can you believe that Steve and L saw, they actually saw, that we literally made adoption papers for Steve?
	U	Did you adopt a child of a different race or culture? Hear the "Parenting in Transracial Adoption" audio: `http://t.co/lFKAps8mCd`
	G	Adoption is a process whereby a person assumes the parenting of another, usually a child, from that person's biological or legal parent or parents, and, in so doing, permanently transfers all rights and responsibilities, along with filiation, from the biological parent or parents.
PHP	Q	RT @taniarascia: NEW Article! How to access JSON data with #php, #javascript, and #jquery! ! `https://t.co/ToS67a4xSA` #webdev #webdevelopmenâĂę
	U	Interested in a #job in #Arlington, VA? This could be a great fit: `https://t.co/hGPzBmGCFw` #PHP #SQL #IT #Hiring `https://t.co/UMiLd88DaS`
	G	At W3Schools you will find complete references of all PHP functions:
Apple cider	Q	Drinking an An Apple A Day by York County Cider - `https://t.co/MqxGhCb0nq`
	U	Apple cider vinegar helps relieve migraines! #acv #vinegar #benefits #health #healthy `https://t.co/UvP72xbNJt`
	G	Apple cider vinegar helps tummy troubles

Table 1: Examples of retrieved documents.

selected verb v is then removed from U and added to V. After n loops, the algorithm returns all selected verbs in V as output.

Algorithm 1 Algorithm for sampling related verbs.

Initialize $U = \{v | v \in D\}$
Initialize $V = \Phi$
for $i = 1 \to n$ **do**
  for $v \in U$ **do**
    Calculate s(v) by Eq. 1
    Calculate r(v) by Eq. 5
    Calculate d(v) by Eq. 7
  end for
  Sample v s.t. $v = \arg\max_{v \in U} \mathrm{s}(v) + \mathrm{r}(v) + \mathrm{d}(v)$
  Move v from U to V
end for
return V

In the third step, we sample the related objective descriptions for each verb. For a selected verb $v \in V$, we extract all objective descriptions of v from D and construct a candidate object set N. Then we sequentially draw the related objective descriptions o from N to construct a ranked object list O[2]. The sampling strategy for a related object o is based on two criteria, i.e. representativeness and diverseness, of o with respect to all objectives in N and all selected objects in O. The calculations of these criteria are illustrated as follows.

The **representativeness of object** o evaluates the degree of averaged similarity of o to all objects in N. The similarity between two objects o and o' referes to the cosine similarity of the vector representation of semantic meanings of these two objects. The calculation is similar to Eq. 1 as

$$\mathrm{sim}(o, o') = \frac{\vec{o} \cdot \vec{o'}}{||\vec{o}|| \times ||\vec{o'}||}, \tag{8}$$

in which $\vec{o}$ and $\vec{o'}$ are the vector representation of semantic meanings of o and o', respectively. The vector representations of object semantic meanings are the mean over the semantic meanings of component words as

$$\vec{o} = \frac{1}{|o|} \sum_{w \in o} \vec{w}, \tag{9}$$

$$\vec{o'} = \frac{1}{|o'|} \sum_{w \in o'} \vec{w}. \tag{10}$$

The representativeness of an object o is calculated as

$$r(o) = \frac{1}{|N|} \sum_{o' \in N} \mathrm{sim}(o, o'), \tag{11}$$

which is similar to the calculation in Eq. 5. Selecting object o with a large representativeness criterion $r(o)$ is equivalent to sampling a representative object for the action v on entity e.

The **diverseness of object** o evaluates the degree of dis-similarity of o to all objects in O, where O is the list of already selected objects. For a candidate object $o \in N$ and a selected object $o' \in O$, we employ the minus of similarity in Eq. 12 to represent their dis-similarity

$$\overline{\mathrm{sim}}(o, o') = -\mathrm{sim}(o, o'). \tag{12}$$

We want the selected object o to be semantically different from already selected objects $o' \in O$. In this sense, we calculate the diverseness of o as

$$\mathrm{d}(o) = \min_{o' \in O} \overline{\mathrm{sim}}(o, o'). \tag{13}$$

The dissimilarity $\overline{\mathrm{sim}}(o, o')$ evaluates the degree of semantic difference between objects o and o'. A large $\overline{\mathrm{sim}}(o, o')$ indicates a huge semantic difference between o and o'. By taking the minimum $\overline{\mathrm{sim}}(o, o')$ in Eq. 13, we can evaluate the semantic difference between the candidate object o and its semantically most similar object in the selected object list O. With a large diverse criterion d(o), we can conclude that the semantic meaning of object o is very different from the semantic meanings of all selected objects in O, and therefore is diverse enough as a candidate for selection.

With the representativeness and diverseness criteria, we are able to sample related objects from N to construct a

[2] Because this process is general for $\forall v \in V$, we do not specify verb v for objects o, N, and O.

ranked object list O for the action v on entity e. The sampling algorithm is depicted in Algorithm 2. At the beginning of Algorithm 2, we initilalize N to be the set of all objective descriptions of v in D, and initialize O to be an empty set. To select m related objectives, we repeat the sampling procedure for m loops. In each loop, we calculate the representativeness and diverseness criteria for each object $o \in N$, and sum them up to generate the sampling criterion. Because a related object should have large values in both representativeness and diverseness criteria, the algorithm selects object o with the largest sum of two criteria at the end of this loop. The selected object o is then removed from N and added to O. After m loops, the algorithm returns all selected objects in O as the output.

Algorithm 2 Algorithm for sampling related objects.

```
Initialize N = {o|o ∈ D(v)}
Initialize O = Φ
for i = 1 → m do
  for o ∈ N do
    Calculate r(o) by Eq. 11
    Calculate d(o) by Eq. 13
  end for
  Sample o s.t. o = arg max_{o∈N} r(o) + d(o)
  Move o from N to O
end for
return O
```

4. EXPERIMENT

The Formal Run of the AM subtask consists of 300 entities. For each entity we sequentially sample 100 actions from the online search results. We submit 3 groups of related actions based on 3 different combinations of the online search strategies, which will be discussed below.

With the Dry Run data of 100 entiteis, we evaluate different online search strategies based on the quality of sampled actions. The online search strategies we have considered include previously mentioned (Q) querying the Twitter Search API, (U) querying the Twitter User Search API and the Twitter User Timeline API, (G) querying Google, and another (S) querying the Twitter Streaming API. Search strategy (S) construct a corpus $D^{(S)}$ for entity e by extracting random Tweets in a Twitter Stream which contain the entity name. For each search strategy, we randomly select 20 Dry Run entities and 10 predicted actions for each entity. The predicted actions are then divided into 20 groups, each for a particular entity. We ask 7 people to manually evaluate the relatedness of these actions, and assign "0" for the not related actions, "1" for the sort of related actions, and "2" for the highly related actions.

Our results suggests that strategy (G) outperforms all other strategies, and strategy (S) performs the worst. We further combine the retrieved documents from strategy (Q), (U), and (G), and repeat the previous method to evaluate the relatedness of sampled actions on these combined corpus. Results suggest that the combination strategies (GQ), (QG), and (QUG) outperform the other combination strategies. The order of combination affects the action prediction result because that the front strategy could contribute more retrieved documents into the combined corpus than the back strategies.

We submit 3 groups of Formal Run submissions. i.e. TUA1-AM-0, TUA1-AM-1, TUA1-AM-2, based on the above combined search strategies (GQ), (QG), and (QUG) respectively, and report the nDCG@10, nDCG@20, nERR@10, and nERR@20 scores for the sampled verbs and sampled verb-object pairs. The detailed evaluations are depicted below.

The nDCG (normalized Discounted Cumulative Gain) score evaluates the usefulness of ranked actions. Given an entity e, the nDCG score for a generated verb $v \in V$ measures the usefulness of v based on its position in list V. For the top k verbs in V, the nDCG@k score evaluates the overall usefulness of these verbs as

$$\text{nDCG@}k = \frac{\text{DCG@}k}{\text{idealDCG@}k}, \tag{14}$$

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}. \tag{15}$$

The nERR (normalized Expected Reciprocal Ranking) score evaluates the action ranking quality by considering the position of these generated actions at which a search user would stop. Specifically, nERR evaluates the expectation of the reciprocal of this position as

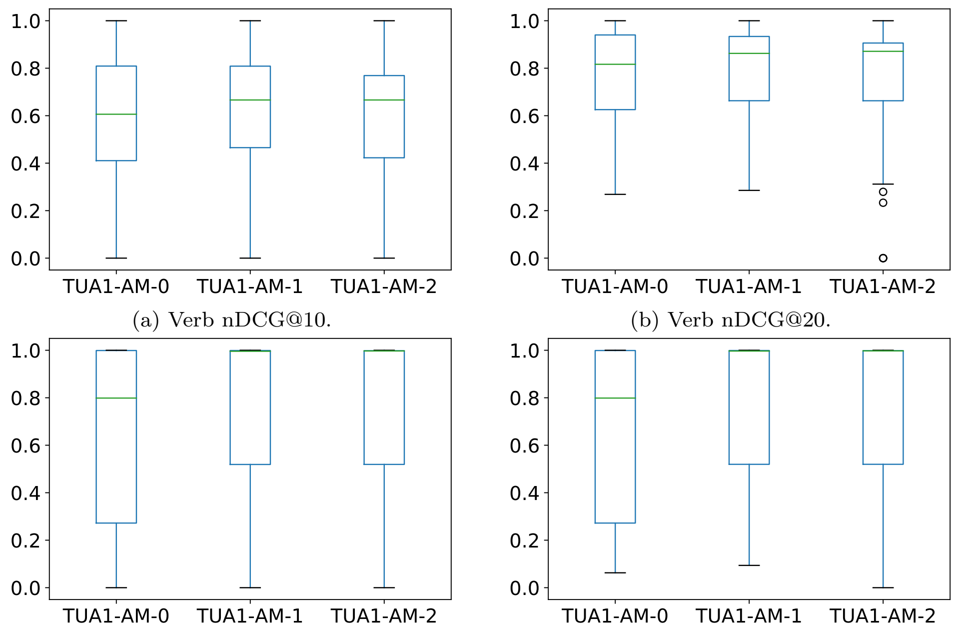
$$\text{nERR@}k = \sum_{i=1}^{k} \frac{1}{i} p(e, a_k) \prod_{j=1}^{i-1} (1 - p(e, a_j)), \tag{16}$$

where e is the query entity and a is a generated action.

Fig. 1 plots the evaluations of sampled verbs for 300 entities in the Formal Run data, with respect to the nDCG@10, nDCG@20, nERR@10, and nERR@20 scores, from 3 groups of submissions. With respect to nDCG@10 and nDCG@20, the verbs in TUA1-AM-1 shows better usefulness than TUA1-AM-0 and TUA1-AM-2, which suggests that the verbs sampled from the (QG) search results are more useful than the verbs sampled from the (GQ) and (QUG) search results. With respect to nERR@10 and nERR@20, the verbs in TUA1-AM-2 show better ranking quality than TUA1-AM-0 and TUA1-AM-1, which suggests that the most related verbs are ranked better, i.e. at the front-most positions in V, from the (QUG) search results than those from the (GQ) and (QG) search results.

The averaged evaluation of verb relatedness, for 3 groups of Formal Run submissions, are shown in Fig. 2. Considering the usefulness of ranked verbs, the TUA-AM-1 group achieves the best average nDCG@10 score 0.6345 and average nDCG@20 score 0.7978. The results are 0.0364 higher in nDCG@10 and 0.1997 higher in nDCG@20 compared to the TUA1-AM-0 group, and 0.0378 higher in nDCG@10 and 0.2011 higher in nDCG@20 compared to the TUA1-AM-2 group. Considering the quality of ranking position of verbs, the TUA1-AM-2 group achieves the best average nERR@10 score 0.7546 and average nERR@20 score 0.7594. The results are 0.0809 higher in nERR@10 and 0.0809 higher in nERR@20 compared to the TUA1-AM-0 group, and 0.0111 higher in nERR@10 and 0.0127 higher in nERR@20 compared to the TUA1-AM-1 group.

Fig. 3 plots the evaluations of sampled verb-object pairs for 300 entities in the Formal Run data, with respect to nDCG@10, nDCG@20, nERR@10, and nERR@20 scores, from 3 groups of submissions. With respect to nDCG@10 and nDCG@20, the verb-object pairs in TUA1-AM-1 shows better usefulness than TUA1-AM-0 and TUA1-AM-2, which



(a) Verb nDCG@10.

(b) Verb nDCG@20.

(c) Verb nERR@10.

(d) Verb nERR@20.

Figure 1: Verb relatedness box plots.

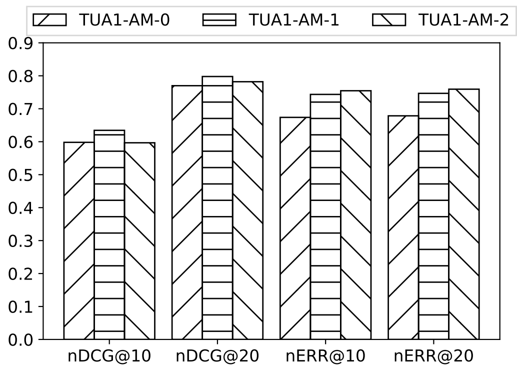


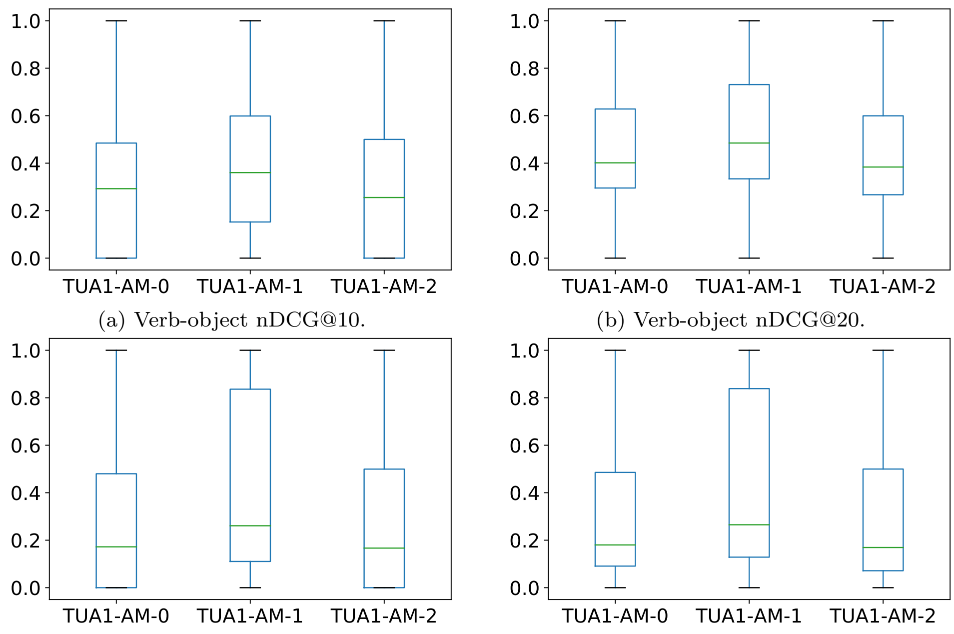
Figure 2: Verb relatedness evaluation.

suggests that the verb-object pairs sampled from the (QG) search results are more useful thant the verb-object pairs sampled from the (GQ) and (QUG) search results. With respect to nERR@10 and nERR@20, the verb-object pairs in TUA1-AM-1 also shows better ranking quality than TUA1-AM-0 and TUA1-AM-2, which suggests that the most related verb-object pairs are ranked better, i.e. at the front-most positions in (V, O), from the (QG) search results than those from the (GQ) and (QUG) search results.

The averaged evaluation of verb-object pair relatedness, for 3 groups of Formal Run submissions, are shown in Fig. 4. Considering the usefulness of ranked verb-object pairs, the TUA-AM-1 group achieves the best average nDCG@10 score 0.3909 and average nDCG@20 score 0.5241. The results are 0.0876 higher in nDCG@10 and 0.2208 higher in nDCG@20 compared to the TUA1-AM-0 group, and 0.0892 higher in nDCG@10 and 0.2224 higher in nDCG@20 compared to the TUA1-AM-2 group. Considering the quality of ranking position of verb-object pairs, the TUA1-AM-1 group also achieves the best average nERR@10 score 0.4047 and average nERR@20 score 0.4172. The results are 0.0993 higher in nERR@10 and 0.0923 higher in nERR@20 compared to the TUA1-AM-0 group, and 0.1016 higher in nERR@10 and 0.0933 higher in nERR@20 compared to the TUA1-AM-2 group.

Finally, we report a case study of the actions generated by our sampling-based algorithm. Table 2 shows some examples of the Formal Run entities and the generated actions with manually annotated scores. The AM subtask annotates the potential actions with scores 0, 1, 2, 3, with the detailed explanations as follows. An action is annotated with score 0 if there is no relevance of the action to the entity. An action is annotated with score 1 if this action can be relevant for the entity. An action is annotated with score 2 if this action has been or will be definitely taken by the entity. An action is annotated with score 3 if some people, organizations or other subjects definitely have taken or will take this action for the entity.

Our submitted actions contains no score 0 actions. We find that most score 1 and score 2 actions might be too personal, such as the action "*started* our measurement unit" from "RT @MsGillsclass: Today we started our measurement unit by brainstorming. #math #gradetwo `https://t.co/JDRvdLIg79`", the action "*got* me crying" from "yo The Sixth Sense got me crying", and the action "*make* a complete use of the other five" from "Sense Money is like a sixth sense without which you cannot make a complete use of the other five". A good number of the score 1 and score 2 actions are extracted from the incorrect sentence parsing results, such as the action "*remove* hair" from "There are also products that come in wipes or liquid form at beauty supply stores to remove hair dye from skin". And some objects in score 2 actions are too short to be fully understood, such as the action *use* Brainstorming from "However, you need to use brainstorming correctly for it to be fully effective".



(a) Verb-object nDCG@10. (b) Verb-object nDCG@20. (c) Verb-object nERR@10. (d) Verb-object nERR@20.

Figure 3: Verb-object relatedness box plots.

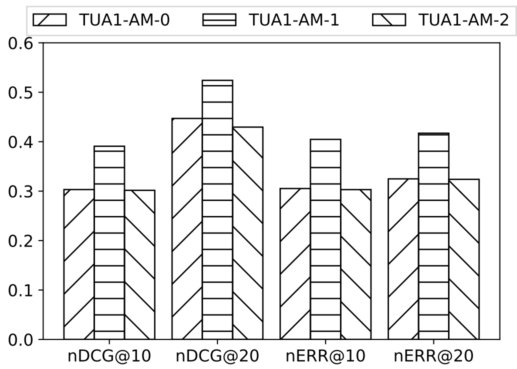


Figure 4: Verb-object relatedness evaluation.

5. CONCLUSION

In this paper, we described our participation in the Action Mining (AM) subtask of the NTCIR-13 Actionable Knowledge Graph (AKG) Task. AM is an open information retrieval problem. To find the most related actions of entities, we propose a sequential sampling method based on online search results. We further propose three criteria, i.e. significance, representativeness, and diverseness to evaluate the relatedness of the candidate actions. In the experiment, we compare the sampled actions from online search results of different strategies. The evaluation of action relatedness on Dry Data suggest that documents retrieved from Google search (G) render better action mining results, while the combination some search strategies, i.e. (GQ), (QG), and (QUG) could generate even better action mining results. We submit three groups of Formal Run submissions, i.e. TUA1-AM-0, TUA1-AM-1, TUA1-AM-2, corresponding to these combined search strategies. The results suggest that action verbs sampled from (QG) search strategy are more useful than the other search strategies, while action verbs sampled from the (QUG) search strategy are randed in the best order. And considering the quality of verb-object pairs, we find that the (QG) search strategy renders the best usefulness and the best ranking property in three groups of submissions.

With a case study, we find some common errors in the sampled actions, such as very personal expressions and incorrect sentence parsing results. Our future work will focus on improving the reasoning of relationship between entities and actions from more general corpus and extracting accurate objects for the entity-related verbs.

Acknowledgments

This research has been partially supported by the Ministry of Education, Science, Sports and Culture of Japan, Grant-in-Aid for Scientific Research(A), 15H01712.

6. REFERENCES

[1] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322. ACM, 2009.

[2] R. Blanco, H. Joho, A. Jatowt, H. Yu, and S. Yamamoto. Overview of ntcir-13 actionable knowledge graph (akg) task. In *Proceedings of the NTCIR-13 Conference*, 2017.

[3] A. Celikyilmaz, D. Hakkani-Tür, and G. Tur. Leveraging web query logs to learn user intent via bayesian latent variable model. 2011.

[4] Idio. Enwiki word2vec model 1000 dimensions. 2015.

[5] D. Jiang, K. W.-T. Leung, and W. Ng. Query intent mining with multiple dimensions of web search data. *World Wide Web*, 19(3):475–497, 2016.

[6] X. Kang, Y. Wu, and Z. Zhang. Learning salient samples and distributed representations for topic-based chinese message polarity classification. *ACL-IJCNLP 2015*, page 68, 2015.

[7] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM, 2008.

[8] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[9] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM Transactions on Information Systems (TOIS)*, 24(3):320–352, 2006.

[10] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 131–138. ACM, 2006.

[11] C.-J. Wang and H.-H. Chen. Intent mining in search query logs for automatic search script generation. *Knowledge and information systems*, 39(3):513–542, 2014.

Entity	Score	Action
Brainstorming	1	started our measurement unit
	1	use twitter”
	2	see “surly curmudgeon” blended so eloquently
	2	use brainstorming
	3	discuss issues
	3	make a brainstorming for cartoonist
Chain smoking	1	smoke why be french know a compulsive heavy smoker
	1	smoke 40-50 cigerettes daily i want
	2	breathe to
	2	start smoking
	3	smoke cigarette
	3	expose to second hand smoke
Hair coloring	1	remove hair
	1	prefer long hair short hair
	2	got ta clean it
	2	remove pubic hair
	3	recommend good home hair colour
	3	got that good hair yo
Language acquisition	1	learn when nothing around be in that language
	1	learn java
	2	improves language acquisition for children aged 0-5 at our talk
	2	find japanese any easier than english, or vice-versa
	3	improve study
	3	learn english
Spoofing attack	1	stop an anxiety attack
	1	overcome panic attack issue
	2	help me setup the csr for our ssl
	2	know who do the spoof
	3	need help with another spoof
	3	know the spoofing for phone nubers when u call rom ur number
The Sixth Sense	1	make a complete use of the other five
	1	watch tv live
	2	got me crying
	2	have a sixth sense
	3	earn money
	3	see dead people” in the sixth sense
Today	1	lose
	1	add hour
	2	get this message
	2	love having you
	3	say be normal how long will that last
	3	get the maximum amount for your refund
Krrish	1	watch tomorrow
	1	make explosion
	2	watch in theatres & feel gauthammenon’s magic
	2	said that it was a good film for children
	3	know the basic story
	3	think about india’s new prime minister mr. narendra modi
Formula One	1	win the formula one driver’s title in 2008
	1	make founder
	2	read more ### filter grand prix*
	2	win the 2006 formula 1
	3	watch formula 1 online
	3	win the f1 this year

Table 2: Entities and the generated actions.