Different Design Choices

TOSEM-2021-0216

To ensure the quality of our proposed approach (i.e., PLAN), we investigate the PLAN's performance under the following design choices: (1) ranking strategy, (2) word embedding technique, and (3) document representation strategy on two API-KI datasets.

1 Effectiveness of Ranking Strategy

ARQ1: How effective is the ranking strategy in retrieving $\langle API, KI \rangle$ pairs?

Different ranking strategies may have different impacts on PLAN. In our work, we use the mutual similarity between the question, potential APIs, and potential results to rank results. We would like to evaluate whether the designed ranking strategy is effective or not. We aim to compare PLAN with the following baseline approaches:

- **PLAN_QR** is the same as PLAN, except in this approach the ranking strategy uses the similarity between the question and potential results.
- **PLAN_QAR** is the same as PLAN, except in this approach the ranking strategy computes the similarity between the question and potential APIs as well as that between potential APIs and potential results.

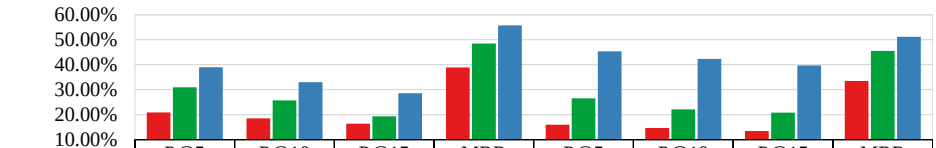
Figure R1 reports the average results of PLAN and the baseline approaches on two AK datasets, respectively. As Figure R1 shown, PLAN outperforms the baseline approaches in terms of P@5 and MRR on two AK datasets. For example, as compared with the baselines, our ranking strategy can improve P@5 and MRR at least by 13.42% and 6.53% on average as compared with the baseline approaches. The p-values are less than 0.02, which confirms that the improvement achieved by the question mapping is statistically significant. In addition, $|\delta|$ values are all greater than 0.59, indicating a large effect size.

We attribute the better performance of PLAN to the following reason: PLAN considers not only the similarity between the question and $\langle API, KI \rangle$ pairs, but also the similarity between the question and the potential APIs as well as that between the potential APIs and the corresponding $\langle API, KI \rangle$ pairs. In this way, our ranking strategy can reflect the possibility between the natural language question and $\langle API, KI \rangle$ pairs effectively. Hence, we use the ranking strategy that computes the mutual similarity to rank results in our work.

2 Effectiveness of Word Embedding Technique

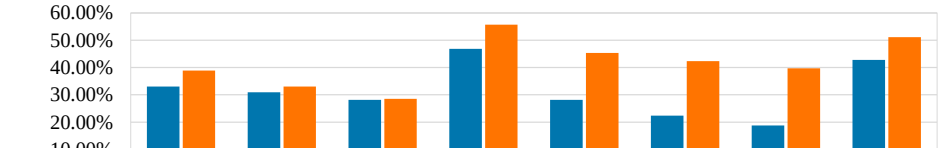
ARQ2: How effective is PLAN under different word embedding techniques?

The existing mainstream word embedding techniques mainly include para2vec (or doc2vec) [1] and word2vec [2]. We would like to verify which technique is more suitable for our PLAN. We compare the P@5, P@10, P@15, and MRR of PLAN (use word2vec to represent the documents) with the following baseline:



	P@5	P@10	P@15	MRR	P@5	P@10	P@15	MRR
	McGill Corpus				Android Corpus			
PLAN_QR	20.87%	18.52%	16.35%	38.85%	16.00%	14.67%	13.44%	33.48%
PLAN_QAR	30.93%	25.73%	19.31%	48.46%	26.50%	22.08%	20.78%	45.48%
PLAN	38.93%	33.00%	28.53%	55.74%	45.33%	42.33%	39.72%	51.15%

Figure R1: Average results of PLAN, PLAN_QR, and PLAN_QAR on two AK datasets



	P@5	P@10	P@15	MRR	P@5	P@10	P@15	MRR
	AK-McGill				AK-Android			
PLAN_P2V	33.07%	30.93%	28.18%	46.89%	28.17%	22.34%	18.78%	42.81%
PLAN	38.93%	33.00%	28.53%	55.74%	45.33%	42.33%	39.72%	51.15%

Figure R2: Average results of PLAN and PLAN_P2V on two AK datasets

- **PLAN_P2V** is the same as PLAN, except this approach uses para2vec to represent the documents.

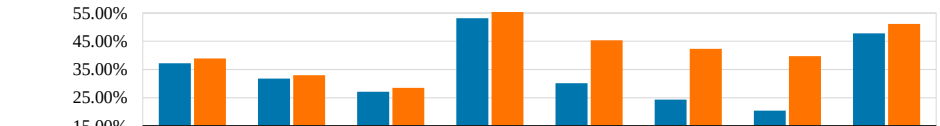
Figure R2 reports the results of PLAN and PLAN_P2V on two AK datasets. Experimental results show that PLAN outperforms PLAN_P2V on two AK datasets, respectively. PLAN improves P@5 and MRR at least by 11.51% and 8.65% on average so that we selected word2vec in our work. We apply a Wilcoxon signed-rank test [3] to evaluate whether the differences between PLAN and PLAN_P2V are statistically significant. The p-values are less than 0.02. The results confirm that the improvement achieved by PLAN is statistically significant. Besides, $|\delta|$ values are all greater than 0.56, indicating a large effect size.

The reason behind PLAN's better performance may be that PLAN needs to use the word embedding to represent not only the documents but also the APIs. For example, PLAN not only uses word embeddings to represent the questions and $\langle API, KI \rangle$ pairs but also potential APIs, then PLAN calculates the mutual similarity between questions, potential APIs and $\langle API, KI \rangle$ pairs. Furthermore, Nguyen et al. [4, 5] and Ye et al. [6] indicated that word2vec is able to project terms of documents and APIs in a shared vector space. Thus, we decide to adopt word2vec to represent the documents in our PLAN.

3 Effectiveness of Document Representation Strategy

ARQ3: How effective is PLAN under different document representation strategies?

Different document representation strategies (i.e., average or concatenate word embeddings of all words in a document) may have impacts on our proposed approach. We would like to investigate the performance of PLAN under different document representation strategies. We compare the P@5, P@10, P@15, and MRR of the following two approaches:



	AK-McGill				AK-Android			
	P@5	P@10	P@15	MRR	P@5	P@10	P@15	MRR
Embedding_avg	37.20%	31.80%	27.11%	53.16%	30.17%	24.34%	20.45%	47.79%
PLAN	38.93%	33.00%	28.53%	55.84%	45.33%	42.33%	39.72%	51.15%

Figure R3: Average Results of Embedding_ave and PLAN on McGill and Android datasets

- **Embedding_ave** is the same as PLAN, except this approach averages the word embeddings of words in a document to represent the document. For each word in a document (e.g., the natural language question, the full API name, and the API description), Embedding_ave first queries the trained word embedding model[1] to get the corresponding word embedding. Then, Embedding_ave averages the word embeddings of all words in each document to represent the document.

Figure R3 shows the average results of Embedding_ave and PLAN on AK-McGill and AK-Android datasets, respectively. As compared with Embedding_ave, PLAN can achieve better performance on P@5, P@10, P@15, and MRR on two datasets, respectively. For example, when comparing with Embedding_ave, PLAN can achieve the average improvement of P@5 and MRR at least by 8.45% and 3.02% on two AK datasets. The p-values are all less than 0.02, indicating that the improvement achieved by PLAN is statistically significant. Besides, $|\delta|$ values are all greater than 0.47, indicating a moderate effect size.

For the better performance of PLAN, this is probably because PLAN is able to preserve the order information of words in the document, while Embedding_ave often loses those order information [1, 7]. For example, PLAN achieves marked improvement on Android datasets as compared with Embedding_avg. This is probably because the length of the content of Android $\langle API, KI \rangle$ pairs is often longer than that of McGill $\langle API, KI \rangle$ pairs. Under this circumstance, PLAN can perform better in preserving the ordering information of words in Android $\langle API, KI \rangle$ pairs than Embedding_ave. To sum up, PLAN outperforms Embedding_ave on two datasets, respectively. Thus, we decide to concatenate the word embeddings of all words in a document to represent the document in our revised version.

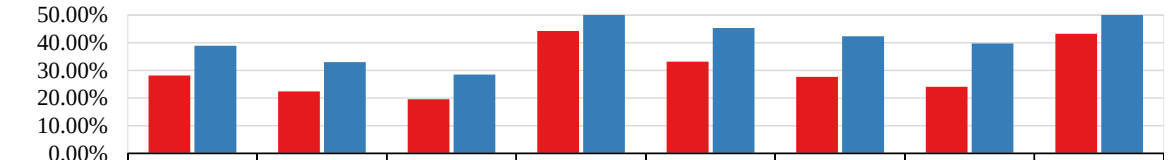
4 Effectiveness of Two Resource-Specific Identification Models

ARQ4: How effective of PLAN when separately using two resource-specific relevance identification models?

Motivation: To find relevant $\langle API, KI \rangle$ pairs from two different resources, one solution is to train two resource-specific relevance identification models (one for $\langle API, FRA \rangle$ pairs and one for $\langle API, QA \rangle$ pairs), respectively. In this RQ, we would like to investigate the performance of PLAN when separately using two resource-specific models.

Approach: We compare the P@5, P@10, P@15, and MRR of the following approach:

[1]We applied the skip-gram model implemented in Gensim [**?**] and the corpus containing 1,733,236 SO posts (see Section 4.1.3 of the paper for details) to train the word embedding model with the default parameters.



	P@5	P@10	P@15	MRR	P@5	P@10	P@15	MRR
	McGill AK dataset				Android AK dataset			
PLAN_tm	28.13%	22.43%	19.53%	44.26%	33.17%	27.67%	24.11%	43.28%
PLAN	38.93%	33.00%	28.53%	55.84%	45.33%	42.33%	39.72%	51.15%

Figure R4: Average results of PLAN_tm and PLAN on two datasets

PLAN_tm (tm stands for **T**wo resource-specific **M**odels) is the same as PLAN, except this approach needs to train two resource-specific relevance identification models (Model_tu for $\langle API, FRA \rangle$ pairs and Model_so for $\langle API, QA \rangle$ pairs). More specifically, PLAN_tm first extracted features of SO posts (i.e., attribute, raw, and co-occurrence features) to train Model_so, and extracted features of API tutorials (i.e., raw, co-occurrence, and extension features) to train Model_tu. Then, PLAN_tm selected relevant $\langle API, FRA \rangle$ pairs and relevant $\langle API, QA \rangle$ pairs to generate relevant $\langle API, KI \rangle$ pairs.

Result: Figure R4 reports the average results of PLAN and PLAN_tm on two datasets. Experimental results show that PLAN outperforms PLAN_tm on two datasets, respectively. For instance, PLAN achieves an average improvement in P@5 and MRR by 11.48% and 9.72% as compared with the PLAN_tm on two datasets. All p-values are smaller than 0.03. The results confirm that the improvement achieved by PLAN is statistically significant. Perhaps, this is because $\langle API, KI \rangle$ pairs in one learning resource may be insufficient to build a good model. Besides, training two models for two resources respectively is a time-consuming process. **Therefore, we decided to train one DTML based relevance identification model, which can find relevant $\langle API, KI \rangle$ pairs from two different resources simultaneously.**

5 Effectiveness of Feature Types

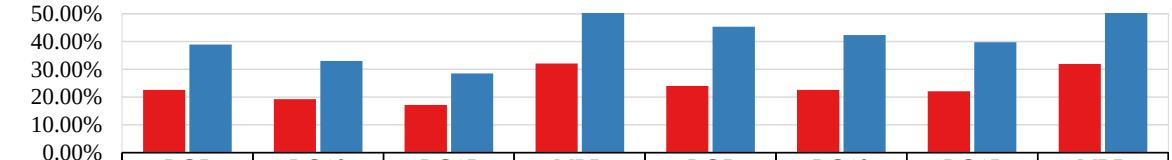
ARQ5: How effective is PLAN when using common features between two resources?

Motivation: PLAN can still retrieve API knowledge if we only use common features between $\langle API, QA \rangle$ pairs and $\langle API, FRA \rangle$ pairs. In this rq, we would like to investigate the effectiveness of common features.

Approach: We compare the P@5, P@10, P@15, and MRR of the following approach:

PLAN_cf (cf stands for **C**ommon **F**eatures) is the same as PLAN, except this approach only uses common features (8 raw features and 5 co-occurrence features) between $\langle API, FRA \rangle$ pairs and $\langle API, QA \rangle$ pairs to train the relevance identification model.

Result: Figure R5 shows the average results of PLAN and PLAN_cf on two AK datasets. From Figure R5, we can observe that PLAN_cf has reasonable performance on two AK datasets, respectively. For example, PLAN_cf achieves P@5 and MRR of 22.53% and 32.09% on McGill datasets. However, PLAN can achieve better performance than PLAN_cf on two



	McGill AK dataset				Android AK dataset			
	P@5	P@10	P@15	MRR	P@5	P@10	P@15	MRR
PLAN_cf	22.53%	19.20%	17.16%	32.09%	24.00%	22.58%	22.11%	31.92%
PLAN	38.93%	33.00%	28.53%	55.84%	45.33%	42.33%	39.72%	51.15%

Figure R5: Average results of PLAN_Cf and PLAN on two datasets

datasets, respectively. For example, PLAN improves P@5 and MRR by 18.87% and 21.49% on average on two datasets when comparing with PLAN_cf. This is probably because using both common features and specific features can capture the commonality and the specificity between $\langle API, FRA \rangle$ pairs and $\langle API, QA \rangle$ pairs. All p-values are smaller than 0.05, indicating that the improvement achieved by PLAN_cf is statistically significant. To sum up, PLAN outperforms PLAN_cf on two datasets, respectively. **Thus, we decide to use both common features and specific features between API tutorials and SO posts in our work.**

References

[1] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Annual Conference on Neural Information Processing Systems*, 2013, pp. 3111–3119.

[3] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[4] T. V. Nguyen, N. M. Tran, H. Phan, T. D. Nguyen, L. H. Truong, A. T. Nguyen, H. A. Nguyen, and T. N. Nguyen, "Complementing global and local contexts in representing API descriptions to improve API retrieval tasks," in *Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 551–562.

[5] T. V. Nguyen, A. T. Nguyen, and T. N. Nguyen, "Characterizing API elements in software documentation with vector representation," in *International Conference on Software Engineering,*, 2016, pp. 749–751.

[6] X. Ye, H. Shen, X. Ma, R. C. Bunescu, and C. Liu, "From word embeddings to document similarities for improved information retrieval in software engineering," in *International Conference on Software Engineering*, 2016, pp. 404–415.

[7] B. Xu, D. Ye, Z. Xing, X. Xia, G. Chen, and S. Li, "Predicting semantically linkable knowledge in developer online forums via convolutional neural network," in *International Conference on Automated Software Engineering*, 2016, pp. 51–62.