We propose Entity Type from Linked Structured Knowledge. Enriching semantics improves performance by involving domain-specific knowledge. Compared to original queries, using the model word embeddings by using application vendor descriptions.

Motivations
- A typical SDS needs a predefined task domain that supports specific functionality; it is not able to dynamically support functions provided by newly installed or not yet installed apps.
- Structured knowledge resources are available (e.g., Freebase, Wikipedia, FrameNet) and may provide semantic information that allows new functionality to be linked into the domain.
- Neural word embeddings can provide semantic knowledge via unsupervised training.

In an open domain, with spoken queries, how can we dynamically and effectively provide the corresponding functions to fulfill users’ requests?

Approaches
1. Generating semantic seeds by using knowledge resources
2. Enriching the semantics with neural word embeddings
3. Retrieving relevant applications or dynamically suggesting users install the applications that support new domain functionality.

Results
- Compared to original queries, using the Freebase knowledge resource (sufficient information about named entities) to extract slot types for enriching semantics of queries achieves 25% and 18% relative improvement of MAP and P@5 respectively.

We enable Extract Type-Embedding-Enriched: Hand-crafted (T)

In an Frame Type of Semantic Parsing
- Retrieving relevant applications or dynamically supporting functions provided by newly installed or not yet installed apps.

- Words with higher similarity suggest that they often occur with common contexts in the embedding training data.

4. Semantics Enrichment

- Main idea: Use distributed word embeddings to obtain the semantically related knowledge for each word.
  1) Model word embeddings by using application vendor descriptions.
  2) Extract the most related words by trained word embeddings for each semantic seed. "text" → "message", "msg".

- Semantic parsing performs well on a generic domain, but cannot recognize domain-specific named entities.

5. Retrieval Process

- Main idea: retrieve the applications that are more likely to support users’ requests via vendor descriptions.
  - Query Refomulation \( Q' \)
    - Embedding-Enriched Query: integrates similar words to all words in \( Q' \)
    - Type-Embedding-Enriched Query: additionally adds similar words to semantic seeds \( S(Q) \)
  - Ranking Model
    \[
    P(Q|A) = \frac{1}{|Q'|} \sum_{x \in Q'} \log P(x|A)
    \]
    - Probability that user speaks \( Q \) to make the request for launching the application \( A \)
    - The application with higher \( P(Q|A) \) is more likely to be able to support the user desired functions.

6. Experiments

- Domain: 13 important application types, accessed the most frequently from Google Play
  - Speech data collected from the users with intents described pictorially (WER = 19.8%)

- Freebase results are better than the embedding-enriched method when \( |Q'| > 50 \), especially for P@5, showing that we can effectively and efficiently expand domain-specific knowledge by types of slots from Freebase.
- Hand-crafted mapping shows that the correct types of slots offer better understanding and tells the room of improvement.