

# Semantic Approaches to Software Component Retrieval with English Queries

Huijing Deng, Grzegorz Chrupała

ETH zürich

ETH Zürich, Tilburg University



## Introduction

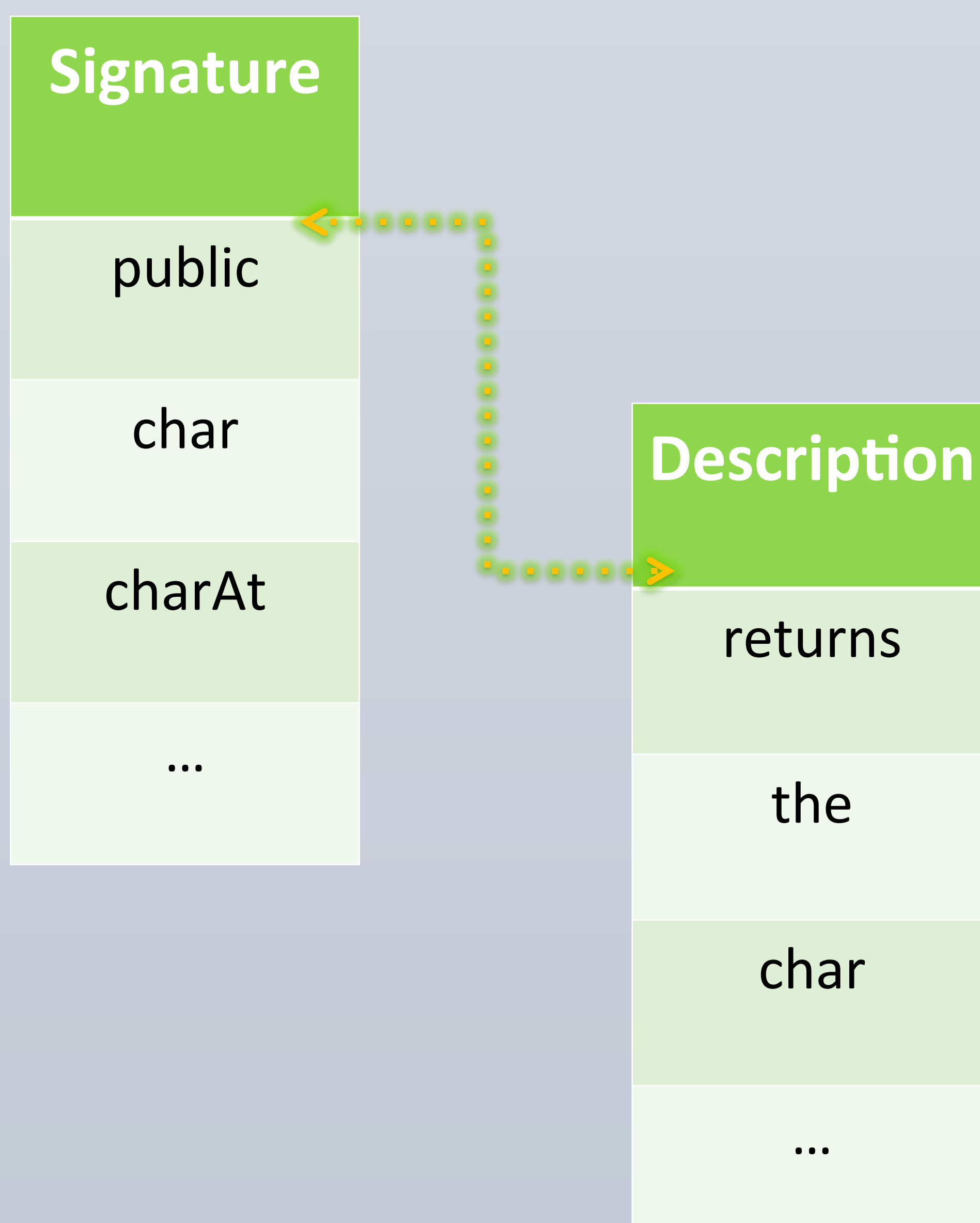
- Enabling code reuse is a major goal in software engineering, and it depends crucially on effective code search interfaces.
- We propose to ground word meanings in source code and use such language-code mappings to make possible a programming library search interface where users can pose queries in English.
- We exploit the fact that there are large programming language libraries which are documented both via formally specified function or method signatures as well as descriptions written in natural language.
- We applied different models to learn the language-signature relation, enabling effective Java methods signature retrieval with English queries.

## Example Data

- Javadoc documentation for method `charAt` in the class `java.lang.String`:

**Class:** `java.lang.String`  
**Signature:** `public char charAt(int index)`  
**Description:** Returns the char value at the specified index. An index ranges from 0 to `length()-1`. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

- We collected such Javadoc documentation of the Java Standard Library.
- To Simulate the queries, we take the first sentence of each description, e.g. “Returns the char value at the specified index”.
- We aim to build a mapping between the terms of Signature and the terms of Description to enable method signatures retrieval using queries formulated in English.



## Models

- To retrieve method signatures using English queries, we rank signatures according to the probability of the method signature  $d$  given query  $q$  (**unigram language model**):

$$p(q|d) = \prod_{w \in q} p(w|d)$$

- **Baseline:** term-matching model, where we set  $p(w|d)$  to the maximum likelihood estimate (MLE) with Jelinek-Mercer smoothing:

$$p(w|d) = (1 - \lambda)f(w|d) + \lambda f(w|D)$$

- $f(w|d)$ : relative frequency of  $w$  in method signature  $d$
- $f(w|D)$ : relative frequency of  $w$  in the whole method collection  $D$

- **PLDA model:**

$$p(q|d) = \prod_{w \in q} \left[ \sum_{t \in T} p(w|t)p(t|d) \right]$$

- $p(w|t)$ : word distribution of topic
- $p(t|d)$ : topic distribution of document
- $T$  represents for all topics

- **Interpolated PLDA model:**

$$p(q|d) = (1 - \alpha) \times p_{plda}(q|d) + \alpha \times p_{baseline}(q|d)$$

- **IBM model:**

$$p(w|d) = (1 - \lambda) \left[ \sum_{u \in d} p(w|u)f(u|d) \right] + \lambda f(w|D)$$

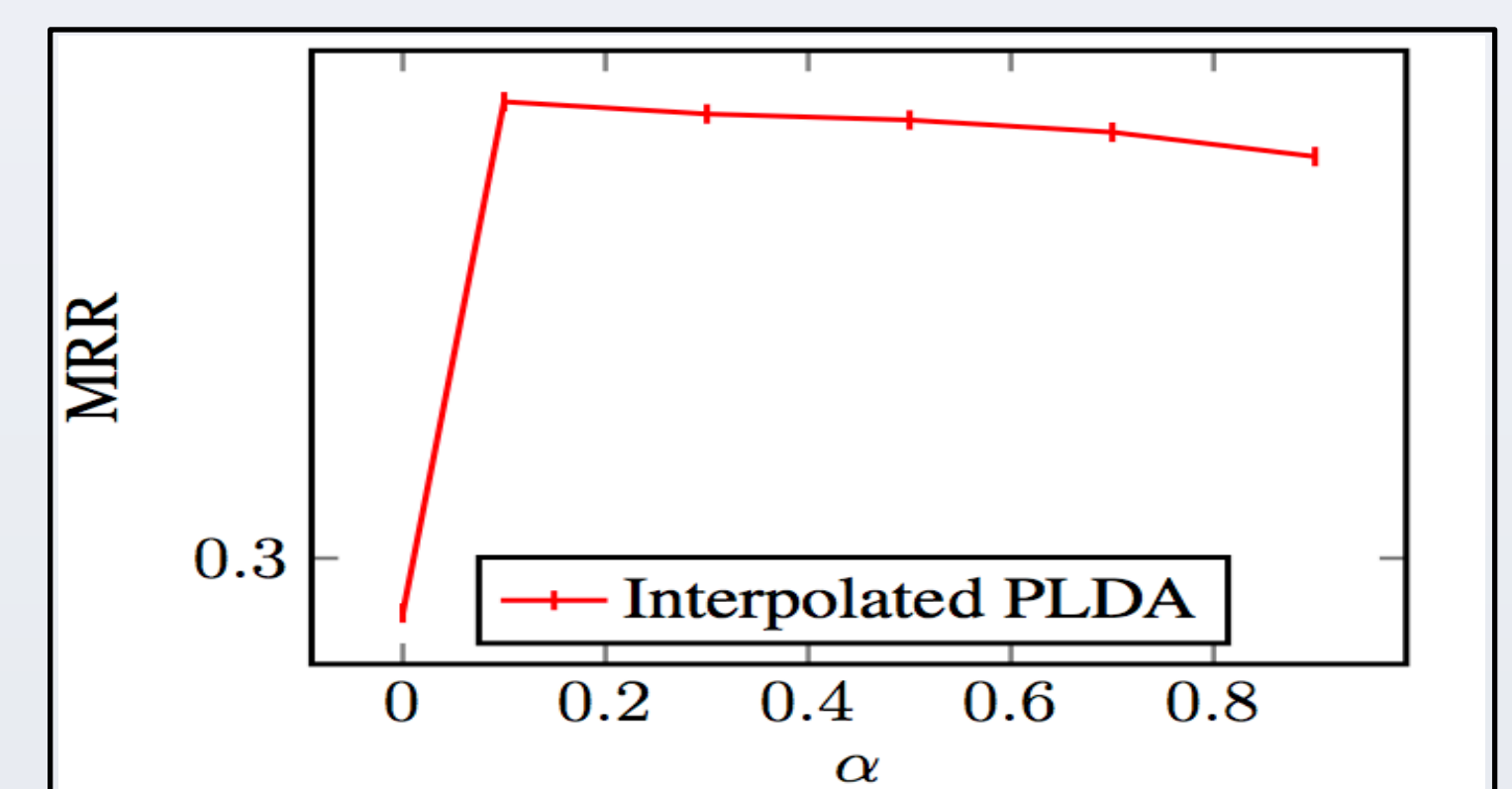
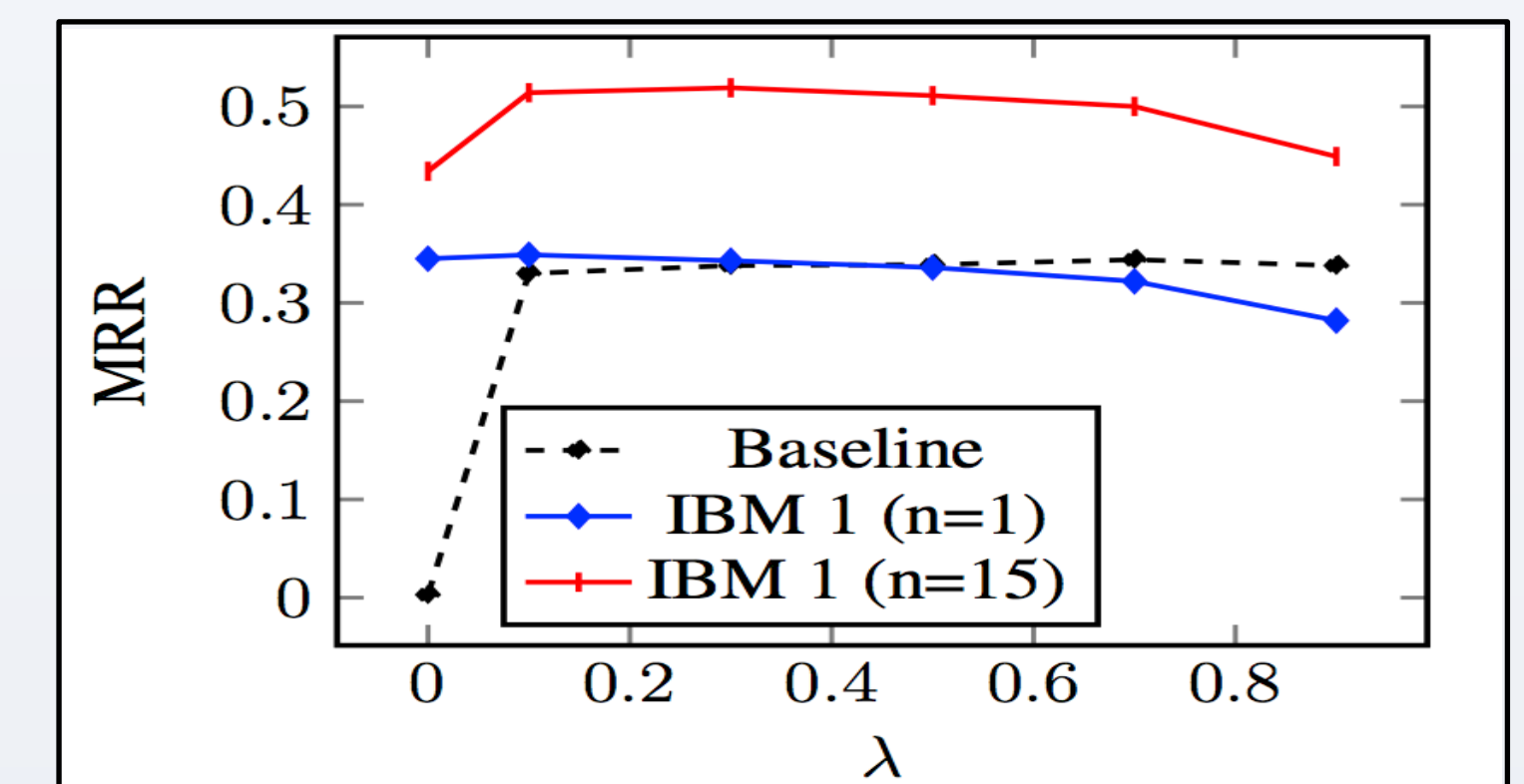
- $p(w|u)$ : probability of translating from term  $u$  to  $w$ ;  $u$  is one term from signature and  $w$  is a term from description
- The translation probability is derived from the parallel corpus of method signatures and descriptions.

## Experiment

- Data: documentation of 6 packages from the Java standard library (Standard Edition 6 API Specification): `io`, `lang`, `math`, `net`, `text`, `util`.
- Split data into training set (60%), validation set (20%), test set (20%).
- To simulate queries, we take the first sentence of descriptions in the test (or validation) set (see Example Data).
- we use mean reciprocal rank (MRR) as our evaluation metric.

## Results

- **Results on validation data**



- **Results on test data**

Model	MRR	Acc@1	Acc@10
Baseline ( $\lambda = 0.7$ )	0.332	0.223	0.530
Interpolated PLDA ( $\alpha = 0.1$ )	0.352	0.242	0.562
IBM model 1 ( $\lambda = 0.3$ )	<b>0.493</b>	<b>0.339</b>	<b>0.793</b>

## Conclusion

- Our research demonstrates that a user-friendly natural language API search interface can be built by exploiting naturally occurring language-code parallel data to ground word meanings.
- The level of performance we have seen is already useful for a real-world application: 80% of queries receive the correct answer within top 10 results.
- Together with this paper we release the data and code developed for this work: [bitbucket.org/schrupala/codeine](https://bitbucket.org/schrupala/codeine).
- For future work, we would like to evaluate our approach on real user queries. It would be also interesting to see how our method generalizes to other programming languages.

## References

- Wei, X. and Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178-185. ACM.
- Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316-321. ACM.
- Berger, A. L. and Lafferty, J. D. (1999). Information Retrieval as Statistical Translation. In *SIGIR*, pages 222-229.