

Army ANT: A Workbench for Innovation in Entity-Oriented Search

José Devezas^[0000-0003-2780-2719] and Sérgio Nunes^[0000-0002-2693-988X]

INESC TEC and Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, s/n, 4200-465, Porto, Portugal
{jld, ssn}@fe.up.pt

Abstract. As entity-oriented search takes the lead in modern search, the need for increasingly flexible tools, capable of motivating innovation in information retrieval research, also becomes more evident. Army ANT is an open source framework that takes a step forward in generalizing information retrieval research, so that modern approaches can be easily integrated in a shared evaluation environment. We present an overview on the system architecture of Army ANT, which has four main abstractions: *(i)* readers, to iterate over text collections, potentially containing associated entities and triples; *(ii)* engines, that implement indexing and searching approaches, supporting different retrieval tasks and ranking functions; *(iii)* databases, to store additional document metadata; and *(iv)* evaluators, to assess retrieval performance for specific tasks and test collections. We also introduce the command line interface and the web interface, presenting a learn mode as a way to explore, analyze and understand representation and retrieval models, through tracing, score component visualization and documentation.

Keywords: Evaluation framework · Entity-oriented search · Representation modeling · Retrieval modeling

1 Introduction

Army ANT is an experimental workbench, built as a centralized codebase for research work in entity-oriented search. Over the years, there have been several experimental frameworks in information retrieval. Some of the most notable include the Lemur Project [1], Terrier [10] and, more recently, Nordlys [8], which is also focused on entity-oriented search. Army ANT was created as a structured framework for testing novel retrieval approaches in a comprehensive manner, even when potentially deviating from traditional paradigms. This required a flexible structure, that we developed by iteratively satisfying the requirements of multiple engine implementations for representing and retrieving combined data [4, Definition 2.3]. An important step in research, that we also motivate and support through our framework, is the continuous documentation of models and collections, which is fundamental for reproducibility, but also useful to advance research, by exploring, learning and building on previous approaches.

2 System architecture

The basic unit of Army ANT is the engine, which must implement the representation model for indexing and the retrieval model for searching. The indexing method has access to one of multiple collection readers and can optionally consider external features. The search method is based on a keyword query, pagination parameters and, optionally, a task identifier, a ranking function and its parameters, and a debug flag. For searching and evaluating over the web interface, each engine is required to have a unique identifier, which frequently describes the representation model and indexed collection (e.g., *lucene-wapo* for a Lucene index over the TREC Washington Post Corpus (WaPo)¹). Each engine has an entry in the YAML configuration file (*config.yaml*), so that it is visible to the web interface. Supported ranking functions, their parameter names and specific values can also be defined in the configuration file. Combinations of selected parameter values can then be used by the evaluation module to launch individual runs, known as evaluation tasks. When completed, each task will provide a performance overview, based on efficiency and effectiveness metrics for each parameter configuration, as well as complementary visualizations and a ZIP archive with intermediate results. Intermediate results include elements like the average precisions for each topic, used in the calculation of the mean average precision, or the results for each individual topic, along with the relevance per retrieved item, according to a ground truth (e.g., qrels from TREC or INEX). This means that, even if Army ANT evolves and no backward compatibility is maintained, the archive can still be downloaded and independently used to compute other metrics, such as statistical tests, or to correct any wrong calculations. Additionally, an overall table, comparing the performance among different runs, is also available for download as a CSV or L^AT_EX file.

Out-of-the-box, Army ANT² provides reader implementations for INEX 2009 Wikipedia Collection [12], TREC Washington Post Corpus, and Living Labs API [3] documents. It also provides a Lucene baseline engine, supporting TF-IDF, BM25 and divergence from randomness, a Lucene features helper, to index and combine external features using the sigmoid approach by Craswell et al. [5], a TensorFlow Ranking [11] engine, which uses Lucene to compute features, and other experimental engines, such as graph-of-entity [6] and hypergraph-of-entity [7]. The latter model supports several tasks, including ad hoc document retrieval (with entities) [2, Ch.8], ad hoc entity retrieval [2, §3.1], related entity finding [2, §4.4.3] and entity list completion [2, p.91], that are not easily explored through conventional evaluation frameworks with the concept of retrieval task. Finally, evaluators are available for the INEX Ad Hoc track and the INEX XER track, as well as for the TREC Common Core track and for the Living Labs API team-draft interleaving online evaluation. On a smaller scale, Army ANT also provides several utility functions, covering DBpedia and Wikidata access, as well as statistics for the measurement of rank concordance and

¹ <https://trec.nist.gov/data/wapost/>

² <https://github.com/feup-infolab/army-ant-install/tree/ecir-2020>

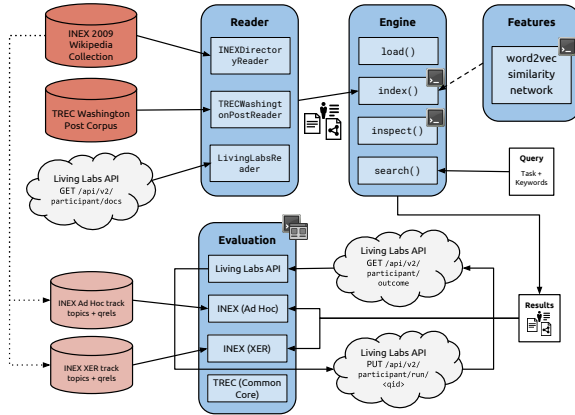


Fig. 1: Army ANT system architecture.

Solid arrows represent information flow, while dashed arrows represent optional interactions. Dotted arrows are simply used to indicate subcomponents of test collections (i.e., topics and relevance judgments).

correlation. Several index inspection, debugging tools and documentation strategies are also integrated into Army ANT’s workflow. The workbench is written in Python, providing integrated implementations for engines written in Java and C++, which we use as examples of cross-language interoperability.

2.1 Overview

We divided the system into what we consider the atomic components of information retrieval research:

1. Iterate over the units of information in a collection (*reader*);
2. Index and search for those units of information (*engine*),
3. Eventually decorate them with additional metadata (*database*);
4. Assess the effectiveness and efficiency of the retrieval (*evaluator*);
5. Obtain as much additional information as possible about the system, in order to reiterate and improve (*web interface* \Rightarrow *learn mode*).

Figure 1 provides an overview of the components in Army ANT, illustrating how they interact with test collections or APIs, as well as with each other. It shows some of the supported implementations, namely readers and evaluators, for both disk-based and REST-based data, and it illustrates feature providers, such as word2vec similarities, that can also be integrated into an index (e.g., providing contextual similarity links to the hypergraph-of-entity). Finally, we can see that a query is defined as a task and a sequence of keywords, and that results can be based on documents, entities, and their relations. Each component may have a command line icon, as well as a web interface icon, showing how it is available to the user.

2.2 Interface

The command line interface can be used for instance for indexing a collection, as seen in Listing 1.1, where the command `index` was issued along with arguments

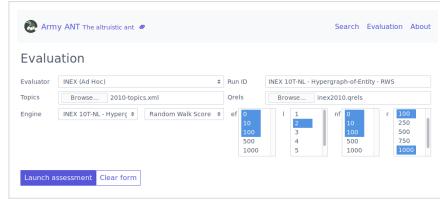


Fig. 2: Evaluation task submission, showing ranking function parameter selection.

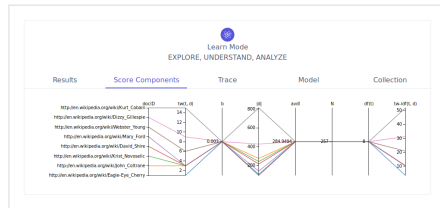


Fig. 3: Learn mode: parallel coordinates visualization of the score components for a query to graph-of-words.

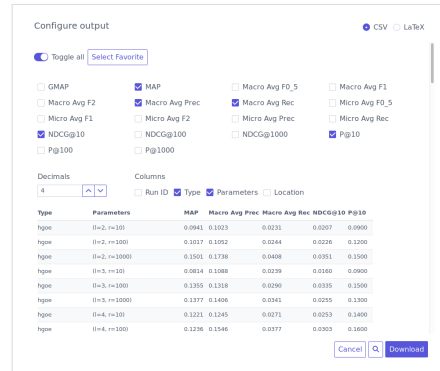


Fig. 4: Exporting evaluation results.

Listing 1.1: CLI indexing example.

```
./army-ant.py index \
--source-path "data/inex-2009" \
--source-reader "inex_dir" \
--index-location "index/lucene" \
--index-type "lucene" \
--db-name "aa_inex"
```

for the source collection, target index and an optional database. A web interface is also available, with modules for accessing search and learn modes, and managing evaluation tasks. Figure 2 illustrates a run for the topics and qrels of the INEX Ad Hoc track, based on the hypergraph-of-entity and the random walk score, configuring values for four parameters. Figure 4 shows the preview dialog for exporting a selection of effectiveness metrics, for all runs. Figure 3 illustrates the score component visualization, a part of the learn mode, which is based on the parallel coordinates system [9].

3 Conclusion

We have presented Army ANT, a flexible workbench for innovation in entity-oriented search and a general platform to support information retrieval research. It promotes reusability by separating collection reading from indexing and by structuring the process of implementing new representation and retrieval models with minimal constraints. One of the biggest strengths of Army ANT is its web interface, where researchers can demo their search engine, as well as explore, understand and analyze several of its facets, either tracing the ranking process for particular queries or visualizing the score components for those same queries. At the same time, we also provide a way for researchers to document their models and collections, using the learn mode to transfer knowledge to other researchers or even to students in a classroom.

Acknowledgements

This work was financed by the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia, through national funds, and co-funded by the FEDER, where applicable. José Devezas is supported by research grant PD/BD/128160/-2016, provided by FCT, within the scope of POCH, supported by the European Social Fund and by national funds from MCTES.

References

1. The Lemur Toolkit for Language Modeling and Information Retrieval | Center for Intelligent Information Retrieval | UMass Amherst. <https://ciir.cs.umass.edu/lemur>, Accessed 20 December 2019
2. Balog, K.: Entity-Oriented Search. Springer (2018)
3. Balog, K., Kelly, L., Schuth, A.: Head first: Living Labs for ad-hoc search evaluation. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management. pp. 1815–1818. CIKM 2014 (2014)
4. Bast, H., Buchhold, B., Haussmann, E., et al.: Semantic search on text and knowledge bases. Foundations and Trends® in Information Retrieval **10**(2-3), 119–271 (2016)
5. Craswell, N., Robertson, S.E., Zaragoza, H., Taylor, M.J.: Relevance weighting for query independent evidence. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 416–423. SIGIR 2005, Salvador, Brazil (Aug 2005)
6. Devezas, J., Lopes, C., Nunes, S.: Graph-of-Entity: A model for combined data representation and retrieval. In: 8th Symposium on Languages, Applications and Technologies. SLATE 2019, Coimbra, Portugal (Jun 2019)
7. Devezas, J., Nunes, S.: Hypergraph-of-entity: A unified representation model for the retrieval of text and knowledge. Open Computer Science Journal **9**(1), 103–127 (Jun 2019)
8. Hasibi, F., Balog, K., Garigliotti, D., Zhang, S.: Nordlys: A toolkit for entity-oriented and semantic search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1289–1292. SIGIR 2017, New York, NY, USA (2017)
9. Inselberg, A.: Parallel coordinates: Visual multidimensional geometry and its applications. In: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval. KDIR 2012, Barcelona, Spain (Oct 2012)
10. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A high performance and scalable information retrieval platform. In: Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval. OSIR 2006, Seattle, Washington, USA (Aug 2006)
11. Pasumarthi, R.K., Bruch, S., Wang, X., Li, C., Bendersky, M., Najork, M., Pfeifer, J., Golbandi, N., Anil, R., Wolf, S.: TF-Ranking: Scalable TensorFlow library for learning-to-rank. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2970–2978. KDD 2019, Anchorage, AK, USA (Aug 2019). <https://doi.org/10.1145/3292500.3330677>
12. Schenkel, R., Suchanek, F., Kasneci, G.: YAWN: A semantically annotated wikipedia XML corpus. In: Datenbanksysteme in Business, Technologie und Web (BTW 2007) – 12. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS). pp. 277–291. Gesellschaft für Informatik e.V., Bonn (2007)