CADET: Computer Assisted Discovery Extraction and Translation

Benjamin Van Durme, Tom Lippincott, Kevin Duh, Deana Burchfield,
Adam Poliak, Cash Costello, Tim Finin, Scott Miller, James Mayfield
Philipp Koehn, Craig Harman, Dawn Lawrie, Chandler May, Max Thomas
Annabelle Carrell, Julianne Chaloux, Tongfei Chen, Alex Comerford
Mark Dredze, Benjamin Glass, Shudong Hao, Patrick Martin, Pushpendre Rastogi
Rashmi Sankepally, Travis Wolfe, Ying-Ying Tran, Ted Zhang
Human Language Technology Center of Excellence, Johns Hopkins University

Abstract

Computer Assisted Discovery Extraction and Translation (CADET) is a workbench for helping knowledge workers find, label, and translate documents of interest. It combines a multitude of analytics together with a flexible environment for customizing the workflow for different users. This open-source framework allows for easy development of new research prototypes using a micro-service architecture based atop Docker and Apache Thrift.[1]

1 Introduction

CADET is an integrated workbench for helping knowledge workers discover, extract, and translate useful information. The user interface (Figure 1) is based on a domain expert starting with a large collection of data, wishing to *discover* the subset that is most salient to their goals, and exporting the results to tools for either *extraction* of specific information of interest or interactive *translation*.

For example, imagine a humanitarian aid worker with a large collection of social media messages obtained in the aftermath of a natural disaster. Her goal is to find messages that contain specific needs, such as hospitals requiring food or medical supplies. She may begin by performing a textual search in our *Discovery* interface (Figure 2). The *Discovery* interface can be customized with different types of search providers, and the aid worker can provide relevance feedback to personalize her search results. After several search sessions, the aid worker may wish to automatically construct a spreadsheet that contains the relevant

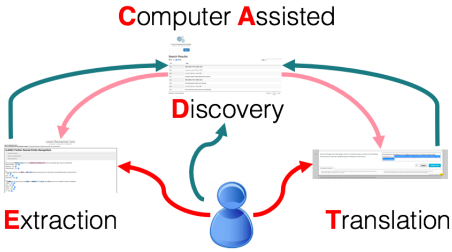

Figure 1: CADET concept

Figure 2: Discovery user interface

information in structured form. To do so, she exports the search results to our *Extraction* interface, where she can provide annotations to help train an information extraction system. The *Extraction* interface allows the user to label any text span using any schema, and also incorporates active learning to complement the discovery process in selecting data to annotate.

Now consider a different user of the same data: suppose a citizen reporter wishes to provide up-to-date news to foreign audiences. He uses our *Discovery* interface to find the messages needed for his story, then exports the text to our *Translation* interface. The *Translation* interface is a computer-assisted translation toolkit that enables him to quickly translate the messages to the target foreign language.

The challenge with building these personalized workflows is that it involves integration of disparate technologies and software components. Individual components like search, information extraction, active learning, and machine translation

[1]Please see http://hltcoe.github.io/cadet for more information. The demo consists of a running system (both online and locally on a laptop), as well as pointers for building the software.

may already be open-sourced and extensible, but to make all these components talk to each other requires a non-trivial amount of effort. CADET is a prototyping framework that demonstrates how this integration can be made easier using a microservice architecture built on Docker and Thrift.[2]

2 Architecture Design

Data serialization In order for analytics to be easily integrated, we developed CONCRETE[3], a data serialization format for Human Language Technology (HLT). It replaces ad-hoc XML, CSV, or programming language-specific serialization as a way of storing document- and sentence- level annotations. CONCRETE is based on Apache Thrift and thus works cross-platform and in almost all popular programming languages, including Javascript, C++, Java, and Python. All analytics in CADET, such as search providers or information extraction systems, are required to be "concrete compliant".

Microservices Analytics within CADET are implemented using a microservice architecture and served up as Docker containers[4]. This allows for rapid prototyping without worrying about the various underlying programming languages and library dependencies. The containers talk via a common CONCRETE microservice API. CADET consists of a set of these microservices that provide functionality for fetching and storing documents, searching, annotating, and training. These service definitions then support code generation of clients and servers in a wide range of languages including Python, Java, C++, Perl, and JavaScript. The decoupled microservice design combined with Thrift's code generation allows researchers to rapidly integrate their own HLT components or compose workflows from existing components.

Platforms CADET workflows have been run in a variety of environments: from standalone laptops with no connection to the internet, to the grid environment, to demonstration systems hosted on Amazon Web Services. Data storage is abstracted, with implementations supporting a simple file-backed directory structure, up to a network distributed Apache Accumulo instance. Each component is a Docker container which can be downloaded, run, and mixed-and-matched based on the need. This framework was used as the basis of a popular course at JHU, with undergraduate students cloning entire frameworks, developing both on laptops and on AWS for projects on knowledge discovery in text.[5]

3 Discovery

Discovery in CADET is presented to the user as a basic IR interface (Figure 2): a query is entered, results are returned in snippet format in a ranked list. These may be additionally labeled for relevance-feedback, interacted with for visualizing pre-computed HLT annotations stored with CONCRETE, or exported to either translation or extraction services. A major goal in the design of the discovery micro-services was to allow for abstracting a large number of non-traditional IR mechanisms, through a common and simple interface. The CADET admin interface allows for on-the-fly changing out of different discovery service providers, in order to allow for compare and contrast studies in how well one perspective on search may be more beneficial than another to a given user. Current discovery providers include:

Keyword search Our baseline discovery approach is keyword search supported by a module implementing our microservice APIs and using the Lucene information retrieval library.[6] This analytic is aware of the Concrete COMMUNICATION data structure, taking a collection of processed communications and performing standard bag of word indexing using the existing document tokenizations.

Cross-lingual For demonstrating the ease of extending the existing pipelines we have implemented an English-Chinese transliteration engine, which is beneficial in particular to named entity search. Many names are transliterated, i.e. characters of Chinese entities may be spelled out in terms of the Latin alphabet, where for an English-speaking user it may be easier to issue queries using the English transliteration, rather than the

[2] CADET is the result of a 9-week summer workshop at the Johns Hopkins University Human Language Technology Center of Excellence (JHU HLTCOE). A motivation for the workshop was the observation that: *there are many more potential users of HLT, each with their own needs, than there are researchers to customize technology to those needs*. Our goal with CADET, besides the workbench itself, is to demonstrate an approach for rapid prototyping and integration.

[3] http://hltcoe.github.io/concrete

[4] https://www.docker.com

[5] This course has recently won an internal educational award at JHU (Lippincott & Van Durme).

[6] https://lucene.apache.org

original Chinese. We currently support a query transliteration system based on an approach similar to Finch and Sumita (2008).

Question Answering From keywords to natural language sentences, one of the CADET discovery service providers is a fully integrated version of recent JHU work in discriminative IR for question answer passage retrieval (Chen and Van Durme, 2017). This allows users to type in a query and get back individual sentences ranked by their likelihood of answering the question.

Mention search We define *entity mention search* as the task of selecting a mention (name, nominal or pronominal) in a currently viewed document, and returning documents most likely to contain mentions of the same entity. We implemented mention search wrapped on top of KELVIN (Finin et al., 2016) which is a multi-year investment in knowledge base population (KBP) research. This framework processes a provided document collection ahead of time to create a knowledge graph with information about entities and their properties, relations and mentions. Mention queries on a pre-processed document interpreted by the KELVIN search service provider as a lookup against a constructed KB, with the provenance information – mentions across the corpus supporting a given entity – returned for displaying to the user. This is an example of ongoing work at the HLTCOE in recasting KBP as supporting technology for structured information retrieval.

Entity Search We define *entity search* as the task of selecting entities from a KB that are similar to a set of entities queried. We implemented Bayesian Sets algorithm and a neural variational auto-encoder version (in preparation), and a user can switch between them using the admin interface. We also provide "query rationale" to explain why a particular entity was returned by displaying important token features for the entities as well as important mentions associated to each entity.

Topic Search We support discovery through notion of "topic search", where documents are analyzed with a topic model and are ranked according to a minimization of Jensen-Shannon divergence between the document's topic distribution and the inferred topic representation of the keywords of a query. Our topic model (JHU Brightside) is a C library implementing stochastic variational inference (Hoffman et al., 2013) for the latent Dirichlet allocation (LDA) (Blei et al., 2003) among others.

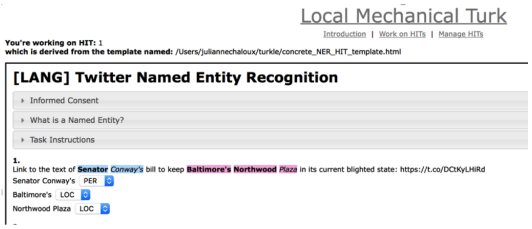

Figure 3: Extraction annotation interface

4 Extraction

After selecting content via the Discovery services, the user can switch to the Extraction web interface to build an information extraction system. CADET provides an interface (Figure 3) for the user to efficiently label this data according to their own sequence tagging schemas, e.g., for named entity recognition (NER).

The extraction annotation framework is also built on CONCRETE, supporting the application of multiple competing systems on content, storing those results together in a single data object, and then visualizing the results back to the user for potential correction. User feedback is stored alongside automatic system results, with annotations from an arbitrary number of users able to stored and easily retrieved later. A user may either correct existing system annotations for later retraining, or existing annotations may be used purely in an active learning service.

Active Learning As CADET is oriented around personalization for a knowledge worker, active learning (AL) is a core consideration, abstracted through a handful of microservice definitions. Interactions between an extraction annotation front-end and learning in the back-end is handled asynchronously by a *data broker*. Content by default is presented to a user for annotation in the rank order provided by a given discovery service, but as a user provides annotations a model may be actively (re-)trained in the background, communicating a preference for new annotations back to the broker, which will reorder subsequent units provided to one or more clients.

Like other services in the CONCRETE stack, the AL service, refered to as LEARN[7], is a specification that can be implemented in any programming language supported by Thrift. The LEARN micro-service's flexibility allows a developer to determine how often the AL server returns a sorted list to the client: a given model implementation

[7] http://hltcoe.github.io/concrete/schema/learn.html

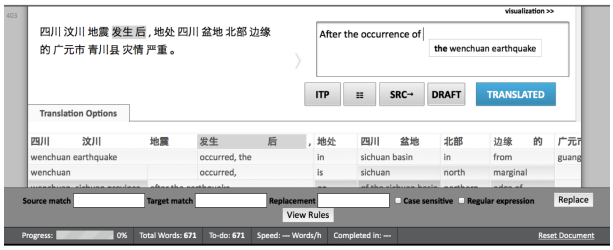

Figure 4: Translation User interface (CASMACAT). The user can see the source sentence (e.g. in Chinese) on the left pane, and can type in or post-edit the translation (e.g. in English) on the right pane.

may be slow to retrain when provided new annotations (such as if retraining in batch mode, from scratch), or be quick to update (such as if supporting incremental training). As the broker moderates the client interaction, then no matter the promptness of the AL backend, the annotator(s) will always have work to do, determined by the most recent update to the annotation ordering.

Information Extraction (IE) systems We have implemented two IE systems under CADET. (1) Pacaya[8] is a Java-based graphical model toolkit (Gormley, 2015) with models that supports tasks such as NER and semantic role labeling. (2) Milner is a sequence-tagging model based on a Hidden Markov Model with Discriminatively Trained Observation Probabilities. This resembles SVM-HMMs, but uses a logistic regression model in place of an SVM for the output. This system was designed foremost for fast training and application on new data.

5 Translation

CADET allows users to export foreign-language documents collected in the discovery phase for translation into e.g. English. We employ CASMACAT (Alabau et al., 2014)[9] (Figure 4) which provides capabilities for: (1) *Post-editing*: first the source sentence is automatically translated by a machine translation system such as Moses (Koehn et al., 2007), followed by a user editing any part of the translated text in order to improve the translation quality; and (2) *Interactive translation prediction*: similar to the auto-complete feature of text input on mobile devices, interactive translation prediction improves translation speed by reducing the number of keystrokes needed in typing. This interface enables the user to create bitext on the discovered documents and improve machine translation in the domain of interest.

[8] https://github.com/mgormley/pacaya-nlp

[9] http://www.casmacat.eu

6 Conclusion

CADET is a framework for rapidly prototyping HLT workflows around a user's needs. We demonstrated a system that enables the knowledge worker to discover material, then actively annotate training data for information extraction or machine translation systems. The underlying principles of our microservice design architecture, based on Docker and Thrift, allows for the combination of a wide variety of analytics. We have built a number of modules that support different microservice definitions such as discovery and active learning, and new developments with different user workflows are planned for the future.

References

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, Germán Sanchis Trilles, and Chara Tsoukala. 2014. Casmacat: A computer-assisted translation workbench. In *EACL*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *JMLR* .

Tongfei Chen and Benjamin Van Durme. 2017. Discriminative Information Retrieval for Question Answering Sentence Selection. In *EACL*.

Andrew Finch and Eiichiro Sumita. 2008. Phrase-based machine transliteration. In *TCAST*.

Tim Finin, Dawn Lawrie, James Mayfield, Paul McNamee, Jessa Laspesa, and Micheal Latman. 2016. HLTCOE participation in TAC KBP 2016: Cold start and EDL. In *Ninth Text Analysis Conference*.

Matthew R. Gormley. 2015. *Graphical Models with Structured Factors, Neural Factors, and Approximation-Aware Training*. Ph.D. thesis, Johns Hopkins University, Baltimore, MD.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *JMLR* .

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.