Dictation Evaluation Reddit Parser

Garcia, Benjamin Lembke, Logan MacMillan, Kyle Smith, Christopher
Stelter, Andrew

September 21, 2018

Contents

1 Introduction to DERP

Dictation Evaluation Reddit Parser – DERP, is designed to be a Domain Specific Language – DSL for interpreting news, or news-like websites. The system was designed as part of a group project for South Dakota School of Mines & Technology's Compilers course.

1.1 What is DERP: Human Queries For a Digital World

Time is something we are regularly lacking and it would be nice to expedite the intake of online news while doing other things such as preparing and eating breakfast, taking a shower, getting dressed, or driving to work. We should be able to say to our phone, "What's going on in the world today?" and get a response containing relevant information that we care about; not just "I'm sorry, I don't understand, 'What's going on in the world today' ". If we want news on a specific topic, such as Tesla, it should be as easy as saying, "What's in the news about Tesla?".

The formats of news intake are largely unchanged in recent years; sure, the internet provides a way for news to be shared, but we still have to hunt down the stories we want, whether we get it through a radio or television broadcast, an online or magazine article, or a newspaper. These broadcasts are rigid and we have no control over what news we hear and there are frequently topics that don't concern us. Articles and newspapers are the same way but we are able to skip what we don't like. The issue with newspapers is we are limited to what the editor decided to include. The issue with articles is that it's down to us to track down relevant stories. A common way to track down interesting articles is an aggregation site such as reddit, steemit, band, and voat to name a few.

1.2 DERP Origins

In the fall of 2018 we were tasked with making a DSL for a class in compiler theory. We discussed many possible topics:

- Constrain/Generate
  - Floor Plans
  - Workout Assistant
  - Computer Part Picker
- Perform a Task
  - Robot "AI" Proposal by Dr. Hinker
  - Simple Image Processing
- Configure a Task
  - News Reader
  - SQL Helper

We decided against a Constrain/Generate model because the ones we came up with all required labeled data which we did not have access to and were not sure we could find or make in a timely manner.

Performing a task seemed like a good idea but the idea of being reliant on getting a robot (in the case of Dr. Hinker's proposal) seemed like a bad idea. We were also unsure of the code that went into it and if we'd be walking into a well documented and fleshed out product or a hack. The image processing was to be something along the lines of simplistic image manipulation.

Lastly we discussed configuration-style tasks. The two proposals were News Reader and an SQL Helper. The basic idea with the news reader is that a user could create custom queries that would obtain news that he/she was interested in while skipping the rest. SQL Helper was an idea to have anyone use SQL. So if a businessman needed some form of report he could ask his secretary to make it and they'd be able to without knowing SQL. The headache of building a system that was robust enough to translate human queries into SQL was deemed too much for the short amount of time we had and so we settled on the News Reader.

We needed a name. Having selected our project we could now determine what to name our team. We settled on Dictation Evaluation Reddit Parser because if we have enough time we'd like it to take spoken words to evaluate. The reddit bit was added because our minimum viable product will be built around reddit.

1.3 Why DERP

DERP allows a developer to create an interface between a person and a feed site such as Reddit, steemit, BAND, and Voat, as described in What is DERP section. This interface allows for a programmer to develop an application that cleanly links an end-user to the feed site(s) of their choice, without the user even being aware that they are using DERP. Without DERP a developer would have to build their own parser to understand what a user wants. DERP allows for a consistent platform for the developer to begin from. Lastly, don't spend time building a solution when one's available for free.

2 DERP Design Goals

2.1 Consolidated Information

The DERP project provides an intuitive way to multiplex multiple news sources into a personalized stream of data. Using DERP, one can specify as many different sources as they want to (provided language plugins are available), and then all of those sources are accessed through the same set of language keywords, regardless of if the source is a subreddit, a news website, or just a file on the user's device. Furthermore, DERP facilitates naming groups of sources and queries to create query macros, allowing users to further personalize what they get out of DERP.

Online news sources regularly expose the same categories of information - date, author, title - DERP acts as an interface for all different article types, allowing users to work only with this high-level information about the articles they are finding. Hiding the details of what exactly is required to find an article that matches a specific criterion allows users to focus more on what they want to read, and less on how they obtain that information.

2.2 Easy to Use Natural English Interface

The second goal of DERP is to design a language that won't feel strange to speak. All of the DERP keywords and syntax are similar to natural English speech patterns. This allows for easy adaptation of DERP into speech-recognition tools such as Google Assistant and Amazon Alexa. Using the natural form of the language, users with one of these devices could speak their program to the interpreter and receive their results immediately.

In addition to being easy to command DERP through natural language, DERP provides output that also feels natural. DERP has a set of phrases available to it for reporting errors or results from user queries. The main output from DERP, articles the user has requested, are outputted as full text so that devices reading them using a text-to-speech system sound as natural as possible.

2.3 Search More, Search Faster

By amalgamating news sources into a personalized feed, DERP allows users to find the information they want with fewer operations. Rather than check each of their favorite subreddits, news sites, and RSS feeds, a user can simply load those sources into DERP and make a general query about them. DERP will do the heavy lifting of finding things the user might be interested, which means the user will be able to spend more of their time actually consuming the information provided and deciding on new topics to get information over.

Because it is an extensible language, DERP users are limited only by the language plugins they use. While some keywords are understood specially by the language, such as those pertaining to dates and times, any language plugin can provide additional fields and keywords, making the language flexible and powerful while keeping the complexity out of the core of the language.

3 The DERP Execution Model

DERP is designed as a language that is easy to use for anyone that wants to get a personalized news feed. To accomplish the goals of this language we needed the language itself to handle a lot of the nuances of the data we will be gathering, as well as allowing the programmer to focus on creating the content that goes into the queries. The programmer needs to worry less on memory, but more on the actual logic of the task at hand. The execution model of DERP will take all of this into account.

3.1 Interactive

The execution of statements within the DERP language will be done with a Read-Eval-Print-Loop (REPL). A statement in our language will be a single line where the delimiter between statements is a newline characters. This REPL will go line by line from typed in text that can either be done manually or through redirected input. For the puposes of creating a prototype this will be done in a text interface similar to the REPL of python and bash. As time permits we will implement this using voice as the input. This will be similar to the REPL, but we will refer to it as the Hear-Eval-Speak-Loop (HESL).

3.2 Hierachical Data (website → boards (subreddits) → posts

DERP is a declaritive language, so the programmer is focused in the logic of the statements to perform queries, otherwise known as selections. This means that the heavy lifting is done by the interpretor that is running the REPL. The interpretor will allow a programmer (user) to use sources such as reddit (if time allows we will add more sources such as cnn, fox, nbc, etc). The interpretor itself will make sure that a statement is valid and conforms to the standars of the current mode and modules loaded for news sources. It then will evaluate the statement and perform a search of the loaded news sources with the criteria specified. Next it will initally pull only enough information for the the user to identify the article's defining information such as its title, date published, and source it is from. The user than can specify follow up statements that may perform summaries, new queries, give details about each article, and so on.

3.2.1 Modes

The interpretor will have 3 modes that depend on the keywords used in the statement. These modes are the main mode, selection mode, and criteria mode.

Main Mode

The DERP interpretor will start in the main mode, also known as the top level mode. This mode allows the user to clear and read statements, as well as enter other modes. The other modes that can be entered are the selection and criteria creation modes. In the main mode the user would need to specify create new criteria or create new selection to enter these modes. To exit the interpretor the user would specify exit in this mode in order to quit out of the interpretor.

Selection Mode

The selection mode would be able to add modules and add and remove queries in the statements, and also be able to save and read the query so far. To exit the mode, the user would specify stop, a keyword, to quit composing selections and enter the main mode again. Examples of using this mode are found in subsubsection 4.1.1.

Criteria Mode

The criteria mode allows for the user to create logical statements that use the add, remove, and save statements. To exit the mode, the user would specify stop, a keyword, to quit composing logical statements and enter the main mode again. Details can be found in subsubsection 4.1.1.

3.2.2 Modules

The interpretor will also allow for modules to be loaded during run time. These modules allow for the user to specify news sources with the news module and searchable criterial with the knowledge module. Examples of module use can be found in subsubsection 4.1.3

News Module

The news source is going to a be a modular component of the interpretor that can be defined by users. This will be a module in python that will define how the interpretor will interface with the source, what are the keywords associated with it, filters and search criteria will also be defined. For example reddit has a Python api that can be interfaced with and will have kewords that get defined such as subreddit and posts. It will also define what is considered a criteria, what a filter can exclude and so on. For simplicity, Reddit will be implemented initally and others added if time permits. When this becomes a HESL, the modules will be built into the interpreter and periodic updates will happen to allow for more sources of news to be suppported.

Knowledge Module

The knowledge modules are used when the user wants to search through articles that were previously fetched using selection statements with news modules. This module would define how articles a are searched through and filtered locally. It would get the contents of the article at this point and filter tham further by the searchable criteria that the user stated.

3.3 Constraints

There are a few constraints on the interpretor implentation for the DERP-Query Language. and this is based around the keywords defined in the subsubsection 4.5.1.
The following must be defined for the main selection mode:

- create
- clear
- read
- exit

The following must be dfined for the selection and criteria modes:

- add
- remove
- clear
- recall
- save as
- read
- stop

Also all modules loaded must must not define their own keywords that are the same as interpretor keywords.

4 The DERP Query Language

The DERP Query Language (DERP-QL) makes it easy to write complex queries for a variety of Internet sources. In DERP, queries are represented by *selections*. Selections find postings from a particular set of sources that match a corresponding set of criteria. The following is an example of a DERP-QL selection.

```
Add posts from subreddit "nba" or subreddit "basketball" on "Lakers"
Remove posts with under 1000 upvotes
Remove posts on "Lebron James"
```

4.1 Main Features

DERP's main features revolve around being composable, interactable, and extensible.

4.1.1 Composable

DERP-QL is composable on several levels. First of all, selections are composed of several *selection statements*. Selections always start with an `Add` statement. `Add` statements always supply at least one source to pull postings from. Additionally, they may restrict the postings to a set of criteria. While selections always start with an `Add` statement, the selection can be further altered by any number of `Add` or `Remove` statements.

Once a selection has been built, it can be saved under a given name. Any saved selection can then be used as a source for another selection. For example, the above example could be split into multiple selections.

```
Create a new selection
    Add posts from subreddit "nba" or subreddit "basketball" on "Lakers"
    Save as "Lakers news"
Stop
Create a new selection
    Add posts from "Lakers news"
    Remove posts with under 1000 upvotes
    Remove posts on "Lebron James"
    Save as "Lakers news without Lebron"
Stop
```

Composing selections makes it easy to assign a name to a number of sources. For example, the first selection in the example immediately above could be broken into two selections.

```
Create a new selection
    Add posts from subreddit "nba" or subreddit "basketball"
    Save as "basketball subreddits"
Stop
Create a new selection
    Add posts from "basketball subreddits" on "Lakers"
    Save as "Lakers news"
Stop
Create a new selection
    Add posts from "Lakers news"
    Remove posts with under 1000 upvotes
    Remove posts on "Lebron James"
    Save as "Lakers news without Lebron"
Stop
```

While reusing selections is handy, it is even more useful to reuse the same filter criteria across multiple sources. While selections must pull from at least one source with an `Add` statement, *criteria* are purely logical constructs. *Criteria* provide a means to compose search criteria without pulling posts from any individual source. The selection above can be rewritten in terms of criteria.

```
Create new criteria
    Add posts on "Lakers"
    Remove posts with under 1000 upvotes
    Remove posts on "Lebron James"
    Save as "Lakers without Lebron"
Stop
Create new selection
    Add posts from subbredit "nba" or subreddit "basketball" \
        matching "Lakers without Lebron"
    Save as "Lakers news without Lebron"
Stop
```

Note: “\” is not a part of DERP-QL. However, “\” here means the statement continues on the next line.

4.1.2 Interactive

DERP-QL is meant to be used in an interactive manner either through a REPL (Read Evaluate Print Loop) or a HESL (Hear Evaluate Speak Loop). DERP-QL specifies keywords and semantics that a DERP-QL REPL/HESL must support. Of particular note are the `Read` and `Recall` keywords. `Read` signals the interpreter to execute a selection. For example, `Read "Lakers news without Lebron"` pauses the interpretation of the containing DERP-QL program, and begins the interpreter dependent handling of the stored selection “Lakers news without Lebron”. The `Read` command may also be used while creating a new selection to check the output of the current selection being built.

`Recall`, on the other hand, makes programming easier for voice users. `Recall` causes the interpreter to list out the statements which make up a given selection. This command is especially helpful in the selection and criteria creation environments when programming by voice.

4.1.3 Extensible

Source modules may define their own grammars which are concatenated into the overall language. For example, the reddit source module defines the `subreddit` keyword. The subreddit keyword takes the next string and uses it to select a portion of reddit to use as a source.

4.2 Primitives

1. Datetime - field type used for criteria based on date fields
2. Number - field type used for criteria on numeric fields
3. Boolean - field type used for criteria on true/false fields
4. String - field type used for criteria on text fields

4.3 Higher Order Types

1. Selection - Composed of one or more selection statements. Contains at least one `Add` statement.
2. Criteria - Composed of one or more criteria statements (boolean expressions). Represents a predicate for whittling down the results from a selection.

4.4 Memory Management

Memory management is automatic, and handled by the Python runtime. Selections, and criteria are persisted to files, and loaded into their representative objects at runtime.

4.5 DERP Operators

4.5.1 String Operators

- with the exact - String comparison equality operator. Includes results where the specified field is an exact match for the provided string.
- like - String comparison fuzzy equality operator. Includes results where the specified field is an approximate match for the provided string.
- in the - Sub-String comparison equality operator. Includes results where a sub-string of the specified field is an exact match for the provided string.

4.5.2 Datetime Operators

- Date on - Date comparison equality operator. Includes results where the specified field is exactly the same date as the provided date.
- date after - Date comparison greater-than operator. Includes results where the specified field is strictly after the provided date.
- date before - Date comparison less-than operator. Includes results where the specified field is strictly before the provided date.

4.5.3 Boolean Operators

- which are | are - Boolean comparison equality operator. Includes results where the specified field contains the boolean value of 'true'.
- which are not | are not - Boolean comparison non-equality operator. Includes results where the specified field contains the boolean value of 'false'.

4.5.4 Numeric Operators

- with exactly | exactly - Number comparison equality operator. Includes results where the specified field contains the same numeric value as the provided numeric value.
- with over | over - Number comparison greater-than operator. Includes results where the specified field contains a value strictly greater than the provided numeric value.
- with under | under - Number comparison less-than operator. Includes results where the specified field contains a value strictly less than the provided numeric value.
- with roughly | roughly - Number comparison epsilon equality operator. Includes results where the specified field contains a value within an epsilon value greater or less than the provided numeric value.

4.5.5 Misc Operators

- matching - Criteria composition operator. The criteria corresponding to the provided name will be textually included into the current criteria or selection.
- from - Selection designation operator. The selection corresponding to the provided name will be textually included into the current selection.
- and | or - Combine the results from the statements on either side of the operator.

4.5.6 Keywords

- exit - Exits the program, and may only be used in the mode selection mode.
- stop - end mode keyword. Ends selection or criteria creation mode and clears the active state.
- clear - reset mode state keyword. Without leaving the current creation mode, clears the active state.
- recall
  - (in create mode) read back the current state. Will read back all statements that have been entered.
  - (in mode selection) read back specified selection or criteria as if it were the active interpreter state.
- save as - store the current selection or criteria with a specified name. A saved selection or criteria may be used in the creation of other selections and criteria.
- read
  - (in create mode) execute the selection in its current state and present the results.
  - (in mode selection) execute the specified selection and present the results.
- add - add a (set of) statements that include or exclude results from a source to a selection or criteria.
- remove - add a (set of) statements that exclude results from one or more sources to a selection or criterias.