Manual Home

Building Magpie

The recommended route for installing Magpie is to download the latest version from Bitbucket and compile it yourself. While this may require installing additional software, it comes with the advantage that it will be much easier to update Magpie as new features are added. This guide will walk you through the installation process.

Required Software

Building Magpie from the latest Bitbucket version requires installing git, the Java Development Kit (JDK), and (optionally) Python with a few libraries.

Linux

If you are running Ubuntu Linux, installing the required software can be accomplished by installing the "git" and "openjdk-7-jdk" packages (although I would recommend "openjdk-8-jdk" or later, if available). There are likely analogous packages on other Linux distributions.

Some features of Magpie require Python with pandas, scikit-learn, numpy, and scipy to function. If you have "pip" installed on your system, you can install these libraries with: `pip install pandas scikit-learn scipy numpy`

Windows

JDK can be installed by downloading and using the installer available from Oracle. After running the installer, verify that the JAVA_HOME environmental variable points to your new installation of JDK. Otherwise, you will get an error about Gradle being unable to find "tools.jar" when trying to compile Magpie.

There are many ways to install Git for Windows. Of these, I would recommend installing the command line version from git-scm.com if you want to follow along with this tutorial. However, there are versions of git with GUIs that integrate with Bitbucket (such as SourceTree), which you might prefer.

The recommended route for installing Python and its required libraries on Windows is to install Conda on your computer. With Conda, You can then install the required packages with: `conda install pandas scikit-learn scipy numpy`

Mac

Installing JDK on OS X can be accomplished by following a guide from Oracle. There is also a guide for installing Git on OS X. As of now, there have not been any issues reported with respect to compiling Magpie on Mac. If you run into problems, please post an issue on the Bitbucket so the developers know to update the guide with a resolution to your issue.

The recommended route for installing the Python dependencies of Magpie, as with Windows, is to use Conda. With Conda, you can then install the required packages with: `conda install pandas scikit-`

```
learn scipy numpy
```

Downloading and Updating Magpie

The source code for Magpie, documentation (including this installation guide), and all of the data required to generate representations for materials is stored in a git repository on Bitbucket. The first step to building Magpie is to clone this repository, which makes it easy to sync with Bitbucket as new changes are made. Cloning can be accomplished by invoking a single command from the command line

```
git clone https://bitbucket.org/wolverton/magpie.git
```

The clone command will create a new directory named "magpie" in the folder in which you called git. This directory contains all of the files shown online at Bitbucket, as well as a folder named ".git" that contains the version history for Magpie and the information needed to connect with Bitbucket and update Magpie. Once the Magpie is cloned, it can be updated again by calling `git pull` from within the Magpie folder. If you call `git log` after updating, you can see descriptions of what was changed and by whom.

Compiling Magpie with Gradle

The preferred method for compiling Magpie is to use Gradle. Gradle is a build tool that automates downloading the libraries required by Magpie, compiling the source code, and other important tasks. You will need an internet connection when running Gradle for it to function properly.

The only step to compiling the JAR file for Magpie is to invoke Gradle by calling `./gradlew jar` from the command line within the Magpie folder. For Windows users, replace "./gradlew" with "gradlew" for this command and all subsequent times Gradle is called. If this is your first time running Gradle on your computer, gradlew will download Gradle itself (gradlew is just a wrapper) and all of the plugins used in the "build.gradle" script provided with Magpie. The "jar" command will cause Gradle to download all of the libraries used by Magpie, compile the code, and create a directory named "dist" that contains Magpie.jar and all of the supporting files. If you update Magpie with `git pull`, you will need to rerun this step to bring Magpie up to date.

To make sure that your copy of Magpie is working properly, call `java -jar dist/Magpie.jar examples/simple-model.in` to run an example script. For Windows users, replace the `/`s with `\`s. If you want to be further sure that Magpie is running properly, run tests for Magpie by calling `./gradlew test`. Gradle will then run several hundred tests that check whether a majority of the methods implemented in Magpie are working as expected. The tests that are most likely to fail are the ones that test the Scikit-learn functionality. If these fail, you may need to either install Scikit-learn on your system or run `make-models.py` in `test-files` to create pkl files compatible with your version of Scikit-learn. If there are other tests that fail, please contact the developers.

The API documentation for Magpie can also be generated with Magpie using Gradle. Calling `./gradlew javadoc` will generate documentation for Magpie in the "javadoc" folder.