bobuilder.*.py

`core.turn_context.py`

- `TurnContext`
  Attributes

  `adapter`: bot adapter for communication
  `activity`: is going on
  `_responded`: can't be set to false
  `_services`: "Map of services and other values cached for the lifetime of the turn."
  `_on_send_activities`,
  `_on_update_activity`,
  `_on_delete_activity`,

`core.dialogs.dialog.py`

- `Dialog(ABC)`
  Dialogue in an `ABC` that has an `id` and `telemetry_client` property, and some pretty intuitive methods. `begin_dialog` starts the dialog. Once started, it may enter a new dialog, after which it may be resumed with `resum_dialog`. The typical interaction will be to `continu_dialog` after a user response. There are also the methods `en_dialog` and `repromp_dialog`. All mentioned methods accept a `DialogContext`, `repromp_dialog` and `en_dialog` take a `TurnContext`, and `resum_dialog` deals with the return and reason for resuming.

elasticsearch

The book *Taming Text* likes java and Apache SOLR, however I'm trying to work on the botbuilder in python and learn about NLP with the NLTK (also python), so something more pythonic seemed in order. I read a nice little article from here `https://solr-vs-elasticsearch.com/`. According to this expert, the most principled reason to choose one over the other is (well, used to be) the use of jsons (more jsons $\Rightarrow$ elasticsearch), mostly due to Python's native and trivial handling of jsons. However, he mentions that elasticsearch (ES) has been diverging from SOLR due to analytics built in. I guess we'll see.

I guess both are based on some Apache product called "Lucene," which "is a high-performance, full-featured text search engine library written entirely in Java." It seems like basic interaction with either one is through HTTP messages, here is an example from the ES documentation:

```
curl -XPUT 'http://localhost:9200/twitter/_doc/1?pretty' \
     -H 'Content-Type: application/json' -d '
{
    "user": "kimchy",
    "post_date": "2009-11-15T13:12:00",
```

```
    "message": "Trying out Elasticsearch, so far so good?"
}'
```

For contrast, here's an example for the SOLR HTTP interface

```
curl "http://localhost:8983/solr/update/extract?&extractOnly=true" \
     -F "myfile=@FILENAME.doc"
```

Note that the `@FILENAME.doc` syntax is curl's way of pointing to a file for content.

There are python bindings for ES, however it's probably more useful to learn about the HTTP interface and then see how the bindings are mapped to it. (That's my plan as of now).