Python notes

Docstring example

```
class Albatross(object):    """A bird with a flight speed exceeding that of an unladen
swallow.
    Attributes:
        flight_speed     The maximum speed that such a bird can attain.
nesting_grounds  The locale where these birds congregate to reproduce.
    """
    flight_speed = 691
    nesting_grounds = "Throatwarbler Man Grove"
```

CRF-suite on AVX2 cpus

using sklearn-crfsuite or python-crfsuite on an AMD system can be very slow due to optimizations in the precompiled wheel files that are specific to intel processors.

I have a branch of python-crfsuite that has flags for avx2 instructions. To see if your cpu supports avx2 instructions, examine the output of `lscpu | grep avx2`. If anything is returned, then your cpu supports the avx2 instruction set.

To use this fork:

Create a virtual environment for this version of python-crfsuite

```
virtualenv ~/venvs/crfsuite
source ~/venvs/crfsuite/bin/activate
```

Clone my fork of python-crfsuite

```
git clone --recurse-submodules git@github.com:ksteimel/python-crfsuite.git
```

Build python-crfsuite

```
python setup.py build
python setup.py install
```

Install additional dependencies

```
pip install sklearn-crfsuite
pip install scikit-learn
```

How much does this help?

Even on intel cpus that support avx2 instructions, the time taken to complete a grid search is reduced. For example, a 5 fold grid search with 10 parameter combinations (e.g. 50 total runs) takes 4 minutes to complete using the version in pypi (on a POS tagging problem in a Turkic language). The avx2 compiled version completes in 3 minutes. This becomes more pronounced as the size of the tagset increases.

Using avx only (e.g. changing -mavx2 to -mavx in the setup.py scipt) still results in improvements to performance. On a pair of intel e5-2680, here are the runtimes for this same benchmark script in minutes. Note that the avx on setting for 16 is having trouble even keeping the cpus loaded because the grid search runs are finishing too fast. This is an issue when the number of tags is small as each task in grid search finishes very quickly and most of the time is spent on task overhead. with longer running tasks, the difference is more noticable.

n_job	avx off	avx on
8	3.5	2.7
16	2.8	2.5