Baby steps in a short-text classification with python

My personal horror story

Alisa Dammer
me: alisadammer.com
@FedorinoGore_90

July 12, 2017

Structure

Initial information collection

Award winning model

Going live

Did I learn anything?

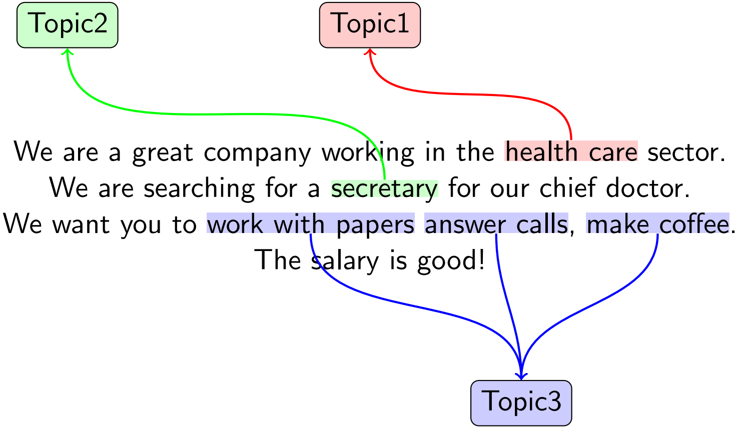
Questions?

What can I do with a text

- Part of the speech tagging
- syntax model
- classification
- text generation
- translation

Binary classification it is!

What can I use?



Topic2

Topic1

We are a great company working in the health care sector.
We are searching for a secretary for our chief doctor.
We want you to work with papers answer calls, make coffee.
The salary is good!

Topic3

KLDB vs ISCO

43412
Informatics, Software development, Assistant/low level complexity

43494
Informatics, Software development, CTO, Tech Lead

Basic tools

- nltk
- sci-kit
- gensim

Evaluation tools

		predicted	
		p	**n**
actual	**p**	True Positive	False Negative
	n	False Positive	True Negative

Let the evaluation begin!

- Bernoulli classification
- Naive Bayesian
- Support Vector Machine
- Decision Tree

Tuning up

- Tweak data set as a whole
- Tweak each item in the data set

Tweaking the item

- Add information
- Remove information
- Stemm the crap out of it

Data transformed!

Some output

```
import nltk.NaiveBayesClassifier as nbc
def build_nb(train):
    modelTrained = nbc.train(train)
    return modelTrained

def train_nb():
    sample = load("path/filename")
    train, test = splitSample(sample, 0.7)
    train = formatForNLTK(train, True, lang)
    test = formatForNLTK(test, True, lang)
    model = build_nb(train)
    getEstimationResults(model, test, labels)
    savePickle("models/classify.pkl", model)
```

Every day we're modelling

```
Time required to train NB is 0.6297673170047347
General TP is  224
General FP is  119
overall accuracy is  0.6530612244897959
confusion matrix is
 [[ 53  32   0]
  [ 16 112   0]
  [  0   0   0]]
```

Doooooom!

Reconnection

- Jython
- Starting python scripts inside of the java code
- Rewrite in Java
- Message brokers
- REST

Deployed with GUnicorn

```
    ...
model = readPickle("model.pkl")
@app.route('/classify', methods=['POST'])
def classify():
    formatted = {}
    results = {}
    if request.method == "POST":
        item, lang = validate(request)
        if lang != expected:
            error_response(lang, model)
        else:
            formatted[model.label] = [item]
            classify(results, formatted, lang, model, model.label)
            logging.info("Classified!")
            return jsonify(results)
```

Is the problem solved?

- Spend more time on base research
- Don't go too deep
- Try graphs first
- Don't be afraid to change the data itself
- Monitoring over historical data
- Have a minimal quality test
- Cross validation is a thing

Thanks for the patience!

Maybe useful information

Tutorials:

- https://pythonprogramming.net/naive-bayes-classifier-nltk-tutorial/
- http://www.nltk.org/book/ch06.html
- http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
- http://scikit-learn.org/stable/modules/svm.html
- http://www.nltk.org/_modules/nltk/metrics/confusionmatrix.html

Basic:

- http://www.linguistics.fi/julkaisut/SKY2006_1/1.6.6.%20NIVRE.pdf
- http://blog.josephwilk.net/projects/latent-semantic-analysis-in-python.html
- https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html
- https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words

Deep:

- https://arxiv.org/pdf/1408.5882v2.pdf
- http://karpathy.github.io/neuralnets/
- http://course.fast.ai/lessons/lesson2.html