Statistical Tools for Linguists

Cohan Sujay Carlos
Aiaioo Labs
Bangalore

Text Analysis and Statistical Methods

- Motivation
- Statistics and Probabilities
- Application to Corpus Linguistics

Motivation

- Human Development is all about Tools
  - Describe the world
  - Explain the world
  - Solve problems in the world
- Some of these tools
  - Language
  - Algorithms
  - Statistics and Probabilities

Motivation – Algorithms for Education Policy

- 300 to 400 million people are illiterate
- If we took 1000 teachers, 100 students per class, and 3 years of teaching per student
  - 12000 years
- If we had 100,000 teachers
  - 120 years

Motivation – Algorithms for Education Policy

- 300 to 400 million people are illiterate
- If we took 1 teacher, 10 students per class, and 3 years of teaching per student.
- Then each student teaches 10 more students.
  - about 30 years
- We could turn the whole world literate in
  - about 34 years

Motivation – Algorithms for Education Policy

Difference:

Policy 1 is $O(n)$ time

Policy 2 is $O(\log n)$ time

Motivation – Statistics for Linguists

We have shown that:

Using a tool from computer science, we can solve a problem in quite another area.

SIMILARLY

Linguists will find statistics to be a handy tool to better understand languages.

Applications of Statistics to Linguistics

- How can **statistics** be useful?
- Can **probabilities** be useful?

Introduction to Aiaioo Labs

- Focus on Text Analysis, NLP, ML, AI
- Applications to business problems
- Team consists of
  - Researchers
    - Cohan
    - Madhulika
    - Sumukh
  - Linguists
  - Engineers
  - Marketing

Applications to Corpus Linguistics

- **What to annotate**
- How to develop insights
- How to annotate
- How much data to annotate
- How to avoid mistakes in using the corpus

Approach to corpus construction

- The problem: ‘word semantics’
- What is better?
  - Wordnet
  - Google terabyte corpus (with annotations?)

Approach to corpus construction

- The problem: ‘word semantics’
- What is better?
  - Wordnet (set of rules about the real world)
  - Google terabyte corpus (real world)

Approach to corpus construction

- The problem: ‘word semantics’
- What is better?
  - Wordnet (not countable)
  - Google terabyte corpus (countable)

For training machine learning algorithms, the latter might be more valuable, just because it is possible to tally up evidence on the latter corpus.

Of course I am simplifying things a lot and I don’t mean that the former is not valuable at all.

Approach to corpus construction

So if you are constructing a corpus on which machine learning methods might be applied, construct your corpus so that you retain as many examples of surface forms as possible.

Applications to Corpus Linguistics

- What to annotate
- **How to develop insights**
- How to annotate
- How much data to annotate
- How to avoid mistakes in using the corpus

Problem : Spelling

1. Field
2. Wield
3. Shield
4. Deceive
5. Receive
6. Ceiling

Courtesy of http://norvig.com/chomsky.html

Rule-based Approach

“I before E except after C”

-- an example of a linguistic insight

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

Probabilistic Statistical Model:

- Count the occurrences of ‘ie’ and ‘ei’ and ‘cie’ and ‘cei’ in a large **corpus**

$P(IE) = 0.0177$

$P(EI) = 0.0046$

$P(CIE) = 0.0014$

$P(CEI) = 0.0005$

Courtesy of http://norvig.com/chomsky.html

Words where ie occur after c

- science
- society
- ancient
- species

Courtesy of http://norvig.com/chomsky.html

But you can go back to a Rule-based Approach

"I before E except after C only if C is not preceded by an S"

-- an example of a linguistic insight

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

What is a probability?

- A number between 0 and 1
- The sum of the probabilities on all outcomes is 1

Heads

Tails

- P(heads) = 0.5
- P(tails) = 0.5

Estimation of P(IE)

P(“IE”) = C(“IE”) / C(all two letter sequences in my corpus)

What is Estimation?

P("UN") = C("UN") / C(all words in my corpus)

Applications to Corpus Linguistics

- What to annotate
- How to develop insights
- **How to annotate**
- How much data to annotate
- How to avoid mistakes in using the corpus

How do you annotate?

- The problem: ‘named entity classification’
- What is better?
  - Per, Org, Loc, Prod, Time
  - Right, Wrong

How do you annotate?

- The problem: ‘named entity classification’
- What is better?
  - Per, Org, Loc, Prod, Time
  - Right, Wrong

It depends on whether you care about precision or recall or both.

What are Precision and Recall

Classification metrics used to compare ML algorithms.

Classification Metrics

Politics

The UN Security Council adopts its first clear condemnation of

Sports

Warwickshire's Clarke equalled the first-class record of seven

How do you compare two ML algorithms?

Classification Quality Metrics

Point of view = Politics

	Gold - Politics	Gold - Sports
Observed - Politics	TP (True Positive)	FP (False Positive)
Observed - Sports	FN (False Negative)	TN (True Negative)

Classification Quality Metrics

Point of view = Sports

	Gold - Politics	Gold - Sports
Observed - Politics	TN (True Negative)	FN (False Positive)
Observed - Sports	FP (False Negative)	TP (True Positive)

Classification Quality Metric - Accuracy

Point of view = Sports

	Gold - Politics	Gold – Sports
Observed - Politics	TN (True Negative)	FN (False Positive)
Observed - Sports	FP (False Negative)	TP (True Positive)

$$\mathrm{A}(M) = \frac{TN + TP}{TN + FP + FN + TP}$$

Metrics for Measuring Classification Quality

Point of View – Class 1

	Gold Class 1	Gold Class 2
Observed Class 1	TP	FP
Observed Class 2	FN	TN

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Great metrics for highly unbalanced corpora!

Metrics for Measuring Classification Quality

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

F-Score = the harmonic mean of Precision and Recall

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F-Score Generalized

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

Precision, Recall, Average, F-Score

	Precision	Recall	Average	F-Score
Classifier 1	50%	50%	50%	50%
Classifier 2	30%	70%	50%	42%
Classifier 3	10%	90%	50%	18%

$$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

What is the sort of classifier that fares worst?

How do you annotate?

So if you are constructing a corpus for a machine learning tool where only precision matters, all you need is a corpus of presumed positives that you mark as right or wrong (or the label and other).

If you need to get good recall as well, you will need a corpus annotated with all the relevant labels.

Applications to Corpus Linguistics

- What to annotate
- How to develop insights
- How to annotate
- **How much data to annotate**
- How to avoid mistakes in using the corpus

How much data should you annotate?

- The problem: ‘named entity classification’
- What is better?
  - 2000 words per category (each of Per, Org, Loc, Prod, Time)
  - 5000 words per category (each of Per, Org, Loc, Prod, Time)

Small Corpus – 4 Fold Cross-Validation

Split	Train Folds	Test Fold
First Run	• 1, 2, 3	• 4
Second Run	• 2, 3, 4	• 1
Third Run	• 3, 4, 1	• 2
Fourth Run	• 4, 1, 2	• 3

Statistical significance in a paper

significance

estimate

variance

Method	Discrimination	%
correct	286/329	87 ± 1.9
no-prior	263/329	80 ± 2.2
no-channel	247/329	75 ± 2.4
neither	172/329	52 ± 2.8

Remember to take Inter-Annotator Agreement into account

How much do you annotate?

So you increase the corpus size till that the error margins drop to a value that the experimenter considers sufficient.

The smaller the error margins, the finer the comparisons the experimenter can make between algorithms.

Applications to Corpus Linguistics

- What to annotate
- How to develop insights
- How to annotate
- How much data to annotate
- **How to avoid mistakes in using the corpus**

Avoid Mistakes

- The problem: ‘train a classifier’
- What is better?
  - Train with all the data that you have, and then test on all the data that you have?
  - Train on half and test on the other half?

Avoid Mistakes

- Training a corpus on a full corpus and then running tests using the same corpus is a bad idea because it is a bit like revealing the questions in the exam before the exam.
- A simple algorithm that can game such a test is a plain memorization algorithm that memorizes all the possible inputs and the corresponding outputs.

Corpus Splits

Split	Percentage
Training	• 60%
Validation	• 20%
Testing	• 20%
Total	• 100%

How do you avoid mistakes?

Do not train a machine learning algorithm on the ‘**testing**’ section of the corpus.

During the development/tuning of the algorithm, do not make any measurements using the ‘testing’ section, or you’re likely to ‘cheat’ on the feature set, and settings. Use the ‘validation’ section for that.

I have seen researchers claim 99.7% accuracy on Indian language POS tagging because they failed to keep the different sections of their corpus sufficiently well separated.