February 15, 2015

Natural Language Models and Interfaces: Assignment Part A, Step 1

Cornelis Boon - 10561145, Markus Pfundstein - 10452397,
Thomas Meijers - 10647023

Abstract

Abstract art uses a visual language of shape, form, color and line to create a composition which may exist with a degree of independence from visual references in the world. Western art had been, from the Renaissance up to the middle of the 19th century, underpinned by the logic of perspective and an attempt to reproduce an illusion of visible reality. The arts of cultures other than the European had become accessible and showed alternative ways of describing visual experience to the artist. By the end of the 19th century many artists felt a need to create a new kind of art which would encompass the fundamental changes taking place in technology, science and philosophy. The sources from which individual artists drew their theoretical arguments were diverse, and reflected the social and intellectual preoccupations in all areas of Western culture at that time.

Abstract art, nonfigurative art, nonobjective art, and nonrepresentational art are loosely related terms. They are similar, but perhaps not of identical meaning.

Abstraction indicates a departure from reality in depiction of imagery in art. This departure from accurate representation can be slight, partial, or complete. Abstraction exists along a continuum. Even art that aims for verisimilitude of the highest degree can be said to be abstract, at least theoretically, since perfect representation is likely to be exceedingly elusive. Artwork which takes liberties, altering for instance color and form in ways that are conspicuous, can be said to be partially abstract. Total abstraction bears no trace of any reference to anything recognizable. In geometric abstraction, for instance, one is unlikely to find references to naturalistic entities. Figurative art and total abstraction are almost mutually exclusive. But figurative and representational (or realistic) art often contains partial abstraction.

Both geometric abstraction and lyrical abstraction are often totally abstract. Among the very numerous art movements that embody partial abstraction would be for instance fauvism in which color is conspicuously and deliberately altered vis-a-vis reality, and cubism, which blatantly alters the forms of the real life entities depicted.[1]

1. Introduction

In this assignment we have built n-grams out of the Austen corpus which can be found `http://www-nlp.stanford.edu/fsnlp/statest/austen.txt`. We have done this in python (see Appendices for details on how to run and the results). From these n-grams, we will extract statistics such as the frequency of words and word sequences. Using these statistics, we have developed a language model that can compute the conditional probability of sequences, the probability of a certain sequence using the Chain rule and finally we also compute probabilities of sequences generated by permutating two sets of words.

2. Problem

2.1. Step 1

A probabilistic approach to language models can make use of n-grams. For this assignment we will create unigrams, bigrams and trigrams.

2.2. Step 2

To calculate the probability of a certain sequence we have built a Markov model using the n-gram statistics. To do this, START and STOP symbols are needed in the text at each start of and end of a sentence respectively. There are also methods needed to calculate conditional probabilities and sequence probabilities, the latter via the power rule.

3. Approach

3.1. Step 1

The main approach to building n-grams out of the corpus is to split the corpus into separate words and then build sequences of length n. To count the frequencies of these n-grams, we use a Counter. Finally we order the results using an ordered dictionary and print the results as well as the sum of the frequencies, which is always equal to the total of n-grams in the corpus. (Not the total of unique n-grams)

3.2. Step 2

Step 2 uses the implementation of step 1 to obtain the n-gram frequencies. These frequencies are then used to calculate conditional and sequential probabilities which are explained below.

For the conditional probabilities of a N-gram, we use the approximation: $P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-N-1}^{n-1})$. This can easily be done by just dividing the N-gram frequency through the N-1 frequency, e.g., for a bigram: $P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$ with C being the count of N gram and the N-1-gram respectively.

For the sequential probability $P(w_1, ..., w_i)$ for a sentence of words $w_1, ..., w_i$, we use the well-known chain rule which is composed of conditional probabilities.

For finding the permutations of a list of words, we use a built-in function provided by the itertools module.

3.3. Step 3

[TODO]

4. Results

Please refer to the appendices B.1, B.2 and B.3 for the results of respectively step 1, 2 and 3.

5. Conclusion

5.1. Step 1

This step required a simple implementation of a n-gram counter combined with a few print statements. For $n = 3$ the Python script takes about three seconds to run and gives the correct output.

5.2. Step 2

Step 2 required an implementation of the chain rule, a few functions that would calculate the conditional probabilities and a function that would compute permutations. The script gives the correct output and runs in few seconds.

Appendices

A. Run instructions

A.1. Step 1

```
usage: a1-step1 [-h] [-corpus INPUT_FILE] [-n N] [-m M]

Assignment A, Step 1

optional arguments:
  -h, --help          show this help message and exit
  -corpus INPUT_FILE  Path to corpus file
  -n N                Length of word-sequences to process (n-grams)
  -m M                Number of n-grams to show in output
To exit: use 'exit', 'quit', or Ctrl-D.
```

A.2. Step 2

```
Assignment A, Step 2

optional arguments:
  -h, --help            show this help message and exit
  -corpus INPUT_FILE    Path to corpus file
  -n N                  Length of word-sequences to process (n-grams) [1,inf]
  -m M                  Number of n-grams to show in output
  -conditional-prob-file COND_FILE
                        file for conditional probabilities
  -sequence-prob-file SEQ_FILE
                        file for sequence probabilities
  -scored-permutations  check permutations
To exit: use 'exit', 'quit', or Ctrl-D.
```

A.3. Step 3

[TODO]

B. Results

B.1. Step 1

B.1.1. 10 most frequent n-gram sequences

m'th most frequent n-gram	n=1	n=2	n=3
1	the 20829	of the 2507	I do not 378
2	to 20042	to be 2233	I am sure 366
3	and 18331	in the 1917	in the world 214
4	of 17949	I am 1366	she could not 202
5	a 11135	of her 1264	would have been 189
6	her 11007	to the 1142	I dare say 174
7	I 10381	it was 1010	a great deal 173
8	was 9409	had been 995	as soon as 173
9	in 9182	she had 978	it would be 171
10	it 7573	to her 964	could not be 155

B.1.2. Sum of all frequencies

- **For n = 1**
  Sum of frequencies = 617091
- **For n = 2**
  Sum of frequencies = 617090
- **For n = 3**
  Sum of frequencies = 617089

B.2. Results Step 2

B.2.1. 10 most frequent bigrams

m'th most frequent bigram	bigram
1	of the 2507
2	to be 2232
3	in the 1917
4	I am 1365
5	of her 1264
6	to the 1142
7	it was 1010
8	had been 995
9	she had 978
10	to her 964

Note that the frequencies have not changed in comparison to step 1 (See appendix B.1). The first entry which is different is the 17^{th} one: START I 784

B.2.2. Output for scored permutations

The scored permutations for set A and B with:
$A = \{$*know, I, opinion, do, be, your, not, may, what*$\}$, and
$B = \{$*I, do, not, know*$\}$

These are the 2 most probable permutations for set A (with their probabilities):

1. ('START I do not know what may be your opinion STOP', $3.661091147270908e - 15$),
2. ('START I do not know what your opinion may be STOP', $3.2217602095983985e - 15$)

These are the 2 most probable permutations for set B (with their probabilities):

1. ('START I do not know STOP', $9.17364947759012e - 07$),
2. ('START I know not do STOP', $5.342809642358792e - 09$)

B.3. Results Step 3

[TODO]

[1] W. Pedia. Abstract art. `http://en.wikipedia.org/wiki/Abstract_art`.