Practice IV

N-gram model

Corpus collection

- Form a team of 3 to 4 people
- Export generated text (carefully selected) of each team member
- Extract messages written by the members and create a corpus of messages for each team member
- Tokenize the content of corpora and create new versions of the corpora (tokenized corpora)

Language model creation

- Using each of the tokenized corpus do the following:
  - Extract bigrams and tigrams
  - Calculate frequency of bigrams and trigrams
  - Calculate the conditional probability of bigrams and trigrams
- Two csv file must be generated, one for each n-gram type
- File must have the following format:
  - For bigrams
    - Term 1, Term 2, frequency of bigram, frequency of context (Term 1), conditional probability of bigram
  - For trigrams
    - Term 1, Term 2, Term 3, Frequency of trigram, frequency of context (bigram Term1 + Term2), probability of trigram
- Content of cvs files must be displayed in descending order of frequency of n-grams

Predictive text

Using the language models do the following:

1. Select the kind of feature to extract (bigram or trigram)
2. Write a word (or two words) to start a sentence
3. Use this word(s) to predict the next word
4. Display the 3 most probable words in order from most likely to least likely and a dot character
5. If you no longer wish to continue the sentence, select the dot character and finish the process.
6. If you wish to continue select one of the words
7. Go back to step 3

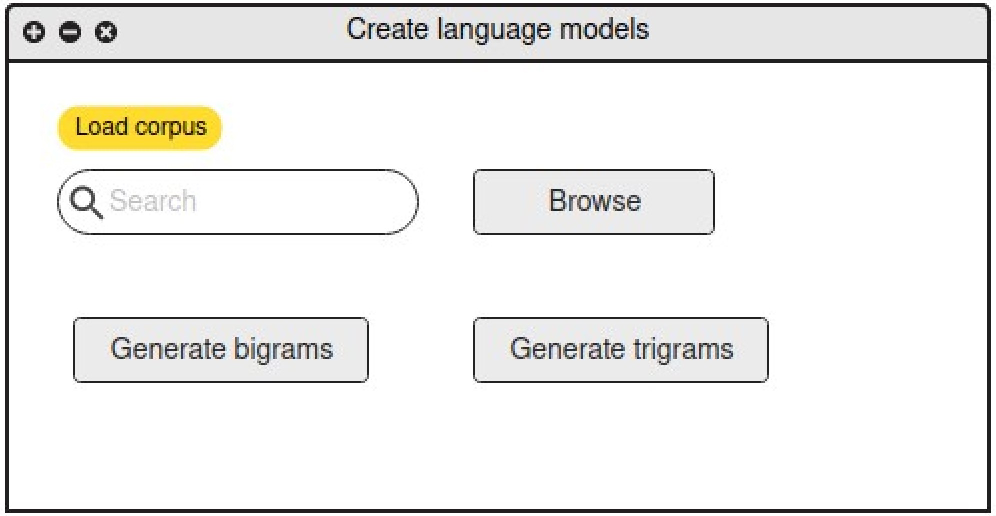
Text generation

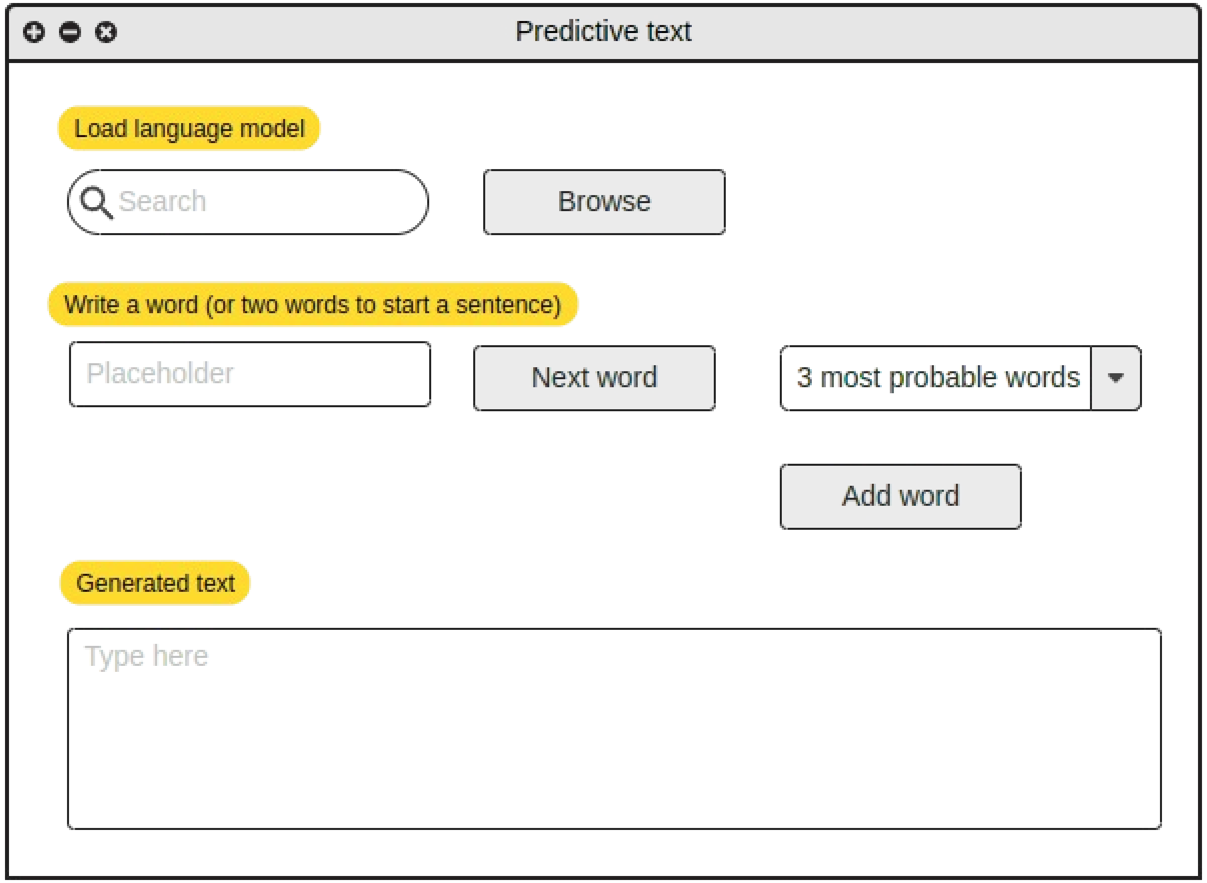
- Using the language models do the following:
  1. Select the kind of feature to use (bigrams or trigrams)
  2. Determine n-grams used to start a sentence
  3. Apply roulette algorithm to select a word that has the n-gram as context
  4. Add the selected word to the sentence
  5. For bigrams use the selected word as the new n-gram context
  6. For trigrams use the last word of the n-gram context and the select word as the new n-gram context
  7. Repeat from step 3 until a character used to finish a sentence is added to the sentence

Conditional probability

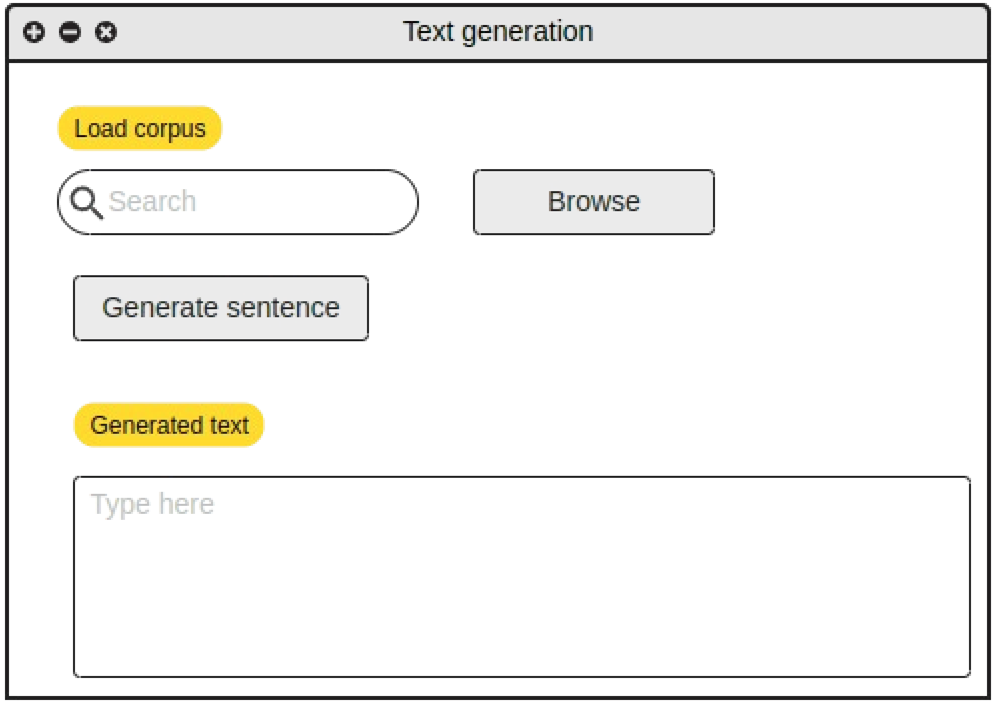
- Using the language models do the following:
  1. Set a test sentence
  2. Select the kind of feature to use (bigrams or trigrams)
  3. Determine the conditional probability of the n-grams (bigrams or trigrams) of the test sentence in each language model. It is important to apply a Laplace smoothing process to handle words not included in the model vocabulary
  4. Determine the joint probability that the test sentence is generated by each of the language models
  5. Show the joint probability of each language model in descending order

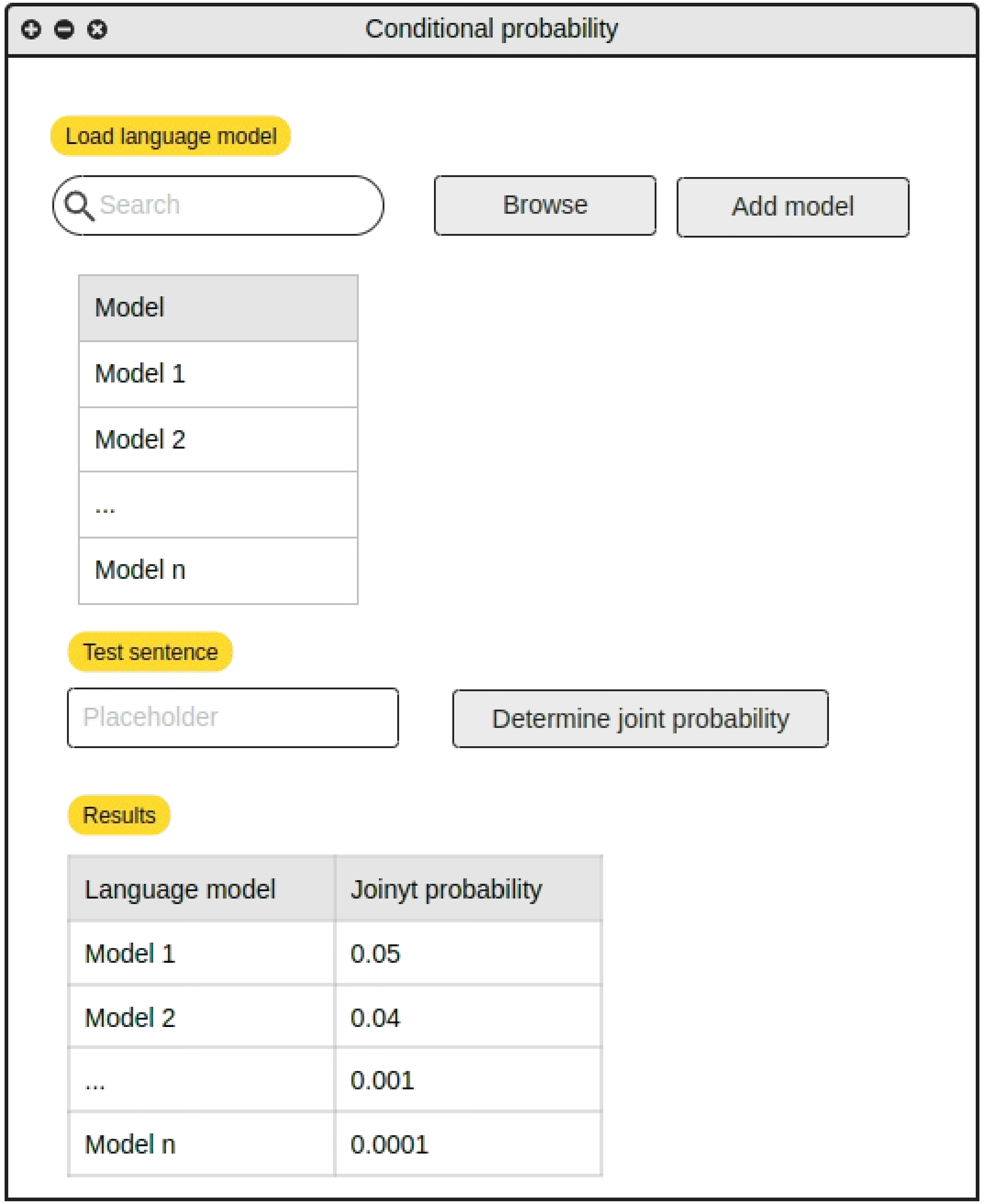
Interface


Create language models
Load corpus
Search
Browse
Generate bigrams
Generate trigrams


Predictive text
Load language model
Search
Browse
Write a word (or two words to start a sentence)
Placeholder
Next word
3 most probable words
Add word
Generated text
Type here

Interface





Model
Model 1
Model 2
...
Model n

Language model	Joinyt probability
Model 1	0.05
Model 2	0.04
...	0.001
Model n	0.0001

Evidence

- Source code
- SVCs files
- Document in PDF with the following content:
  - Introduction. Describe the features of the n-gram model and explain the three performed tasks
  - Development of the language models. Explain the procedure for creating the language models
  - Experiments. Show 5 experiments and results performed for each task
  - Conclusion. Describe the results obtained by the models in the different tasks. Indicate with which features (unigrams or bigrams) and in which tasks the models perform best. Point out the difficulties encountered in the implementation and resolution of the tasks. Indicate what improvements could be made in the future.
- The document must include the names of the team's members
- All the members must upload the evidence