Nepali Language Processing

Function and Non Functional Requirements

Functional Requirements

R0: Word Embeddings Visualization

Description : Word embeddings of nepali words created using various methods like word2vec, embedding layers should be visualized in 2d and 3d graphs and show the required relationship between the words.

R0.0: Visualize word embeddings
Input : Select word embeddings visualization tab
Output : 2d and 3d graphs of the word embeddings vector of few selected words or default 100 or 200 words(Number of words may vary as per requirements).
Processing: All the default word embeddings vectors will be returned and plotted in 2d and 3d graphs.

R0.1: Search words
Input : Keyword to be searched in the graph
Output: Vector of keyword and position in the graphs highlighted
Processing: Keyword searched in the vocabulary and if found its vector is returned and plotted with highlight color in the graph. If not found, display NOT FOUND message. If the keyword is not found in the vocabulary, archive it to the vocabulary database for future references.

R1: Sentimental Classification
Description: When the user enters a sentence and performs sentimental classification, the sentimental classification should be able to distinguish the sentence into positive, negative and neutral with some cutoff probability (about 70%).

R1.1: Classify and result
Input : A sentence
Output: Probabilities of the sentence being positive, negative and neutral.
R1.2: Store the result
Input: A sentence to be classified

Processing: Store the sentence and its predicted label in the sentimental classification table. If the predicted label is incorrect , get the correct label from the user and store it in the database.

R2: Language Modeling
Description: Nepali Language modeling based on both probabilistic and neural approach. Based on the trained language model, it should be able to auto generate the next word given context words.

R2.1: Predict next word
Input : An incomplete sentence
Output : List of words along the probabilities of being the next word.
Processing : The language model should be able to predict the next word from vocabulary based on the previous words provided by the user.

R3: Spelling Correction
Description: Based on the trained language model, the system should suggest correct spelling for a given sentence.
Input: A sentence
Output : Provide suggestions to correct the spelling of the words in the input sentence.
Processing: Firstly, candidate sentences are found using 1 or 2 minimum edit distance. Probability of each candidate sentence and likelihood of error of given sentence is calculated. And suggestions should be made based on maximum posterior.

R4: Interactive web application
User Requirements

-

System Requirement

Non Functional Requirements

R1: Performance and scalability

- The load time for the user interface screens shall take no longer than 3 seconds.
- Queries shall return results within 3 seconds.

R2: Design Constraints

- The system shall be developed using python and postgresql databases.

R3: Standard Compliance

- The graphical user interface shall have a consistent look and feel.

R4: Availability

- The system shall be available all time.

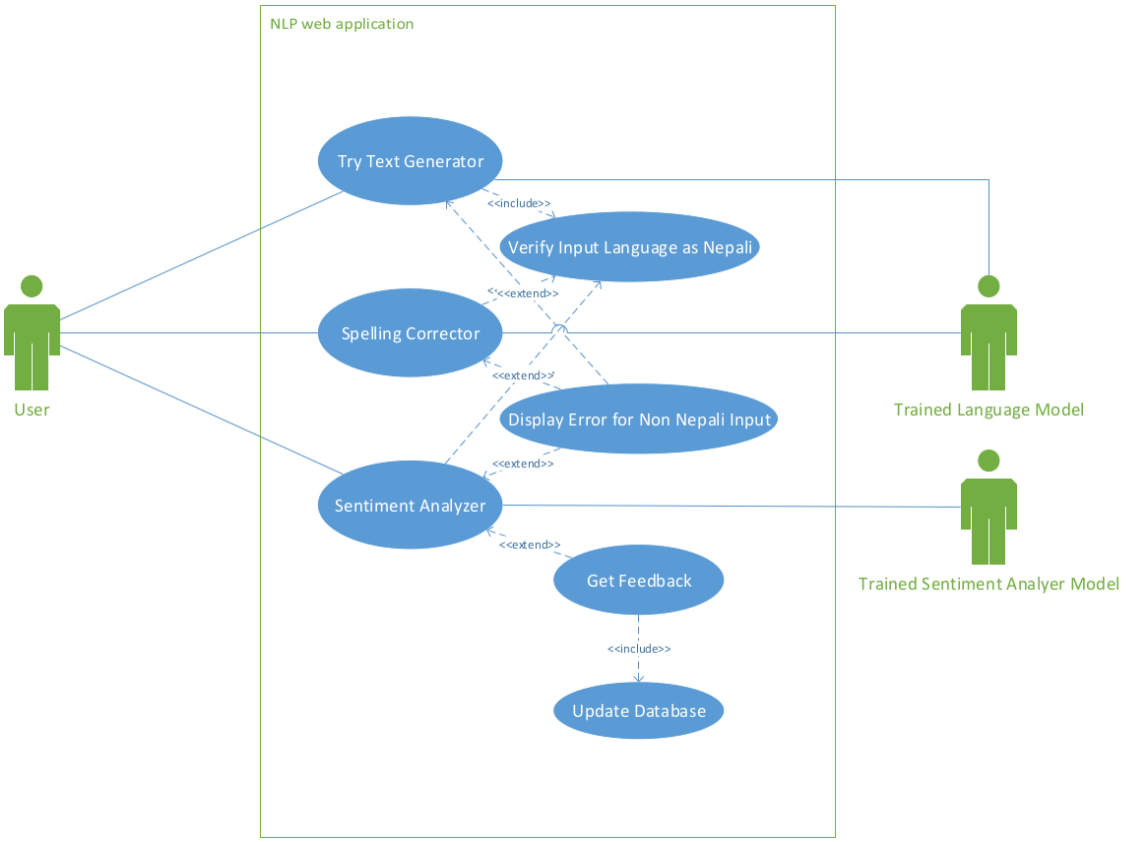
R5: Portability and compatibility

- The system shall be able to run in all browsers.

R6: Reliability, maintainability, availability

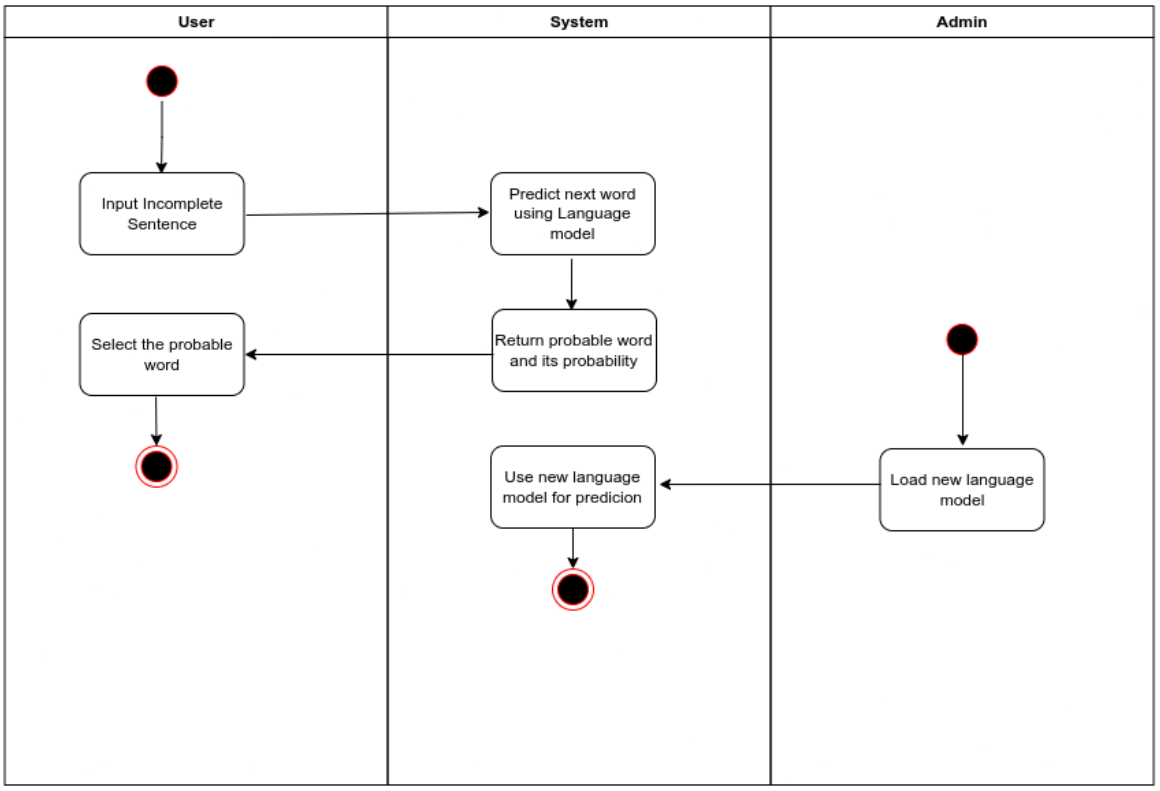
- The mean time to recover from a system failure must not be greater than 2 hours.
- Rate of system failure should be greater than twice a month.

UML use case descriptions and diagrams

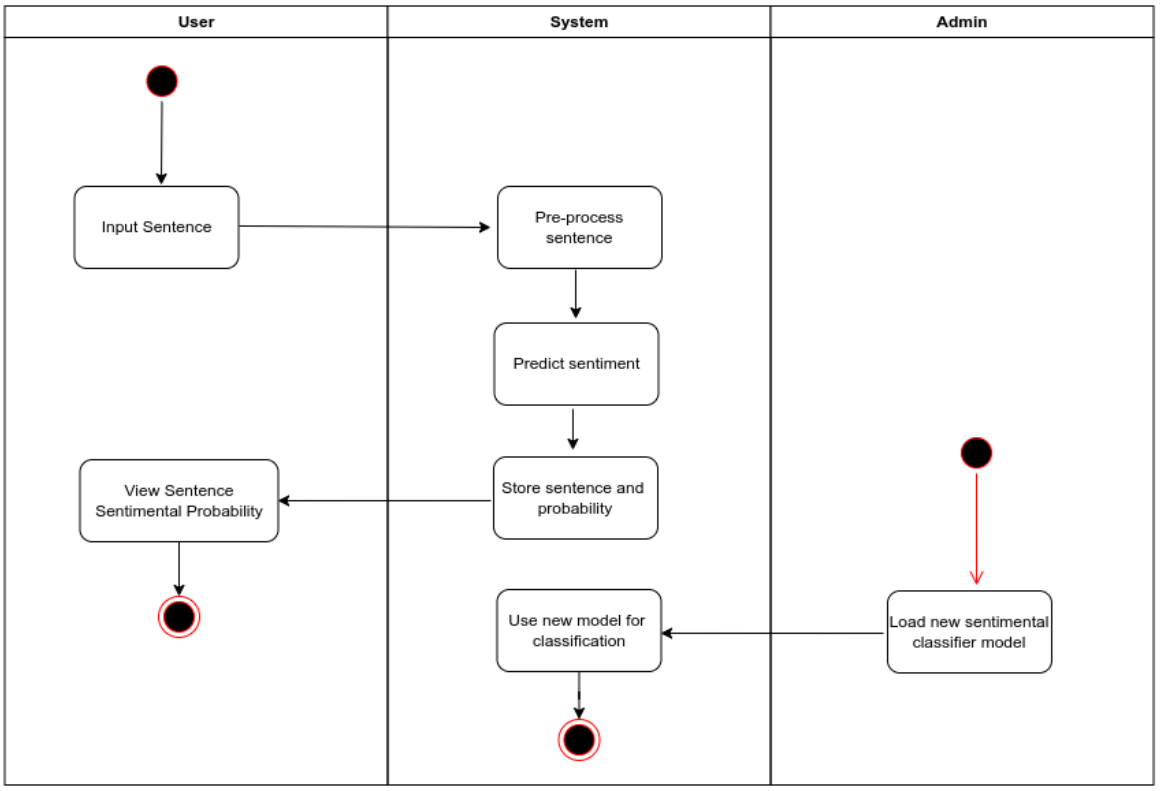


UML Activity Diagram

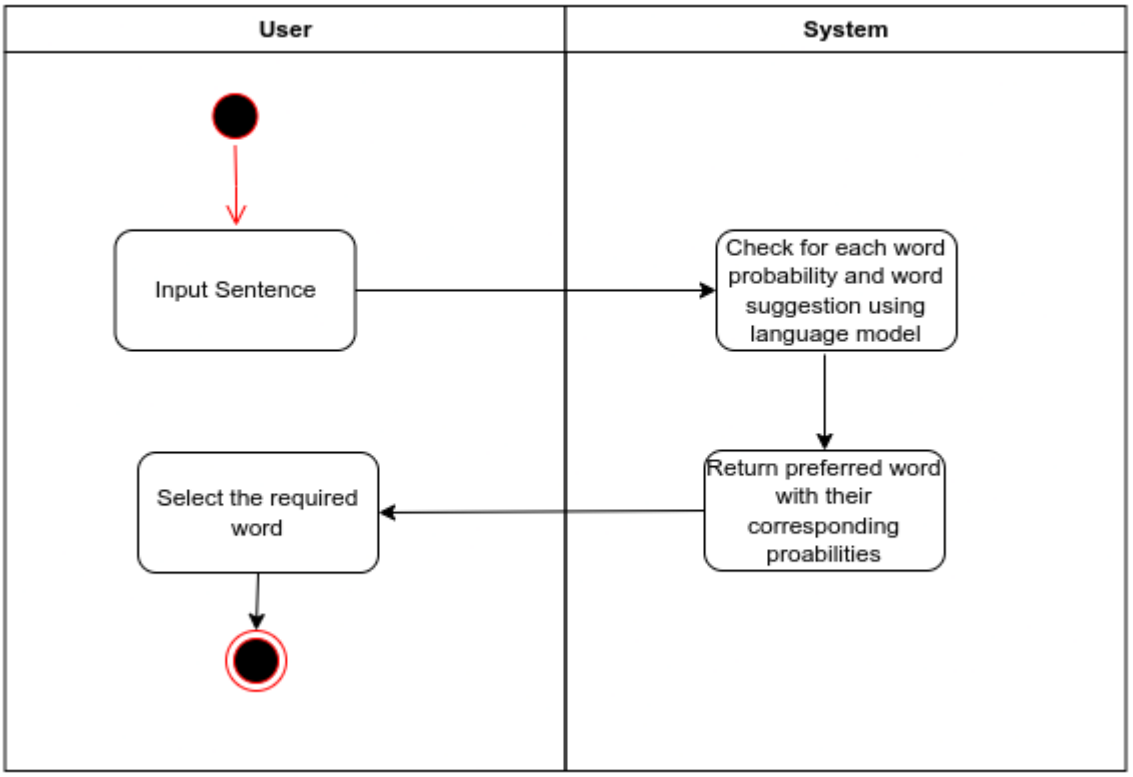
Language model activity diagram



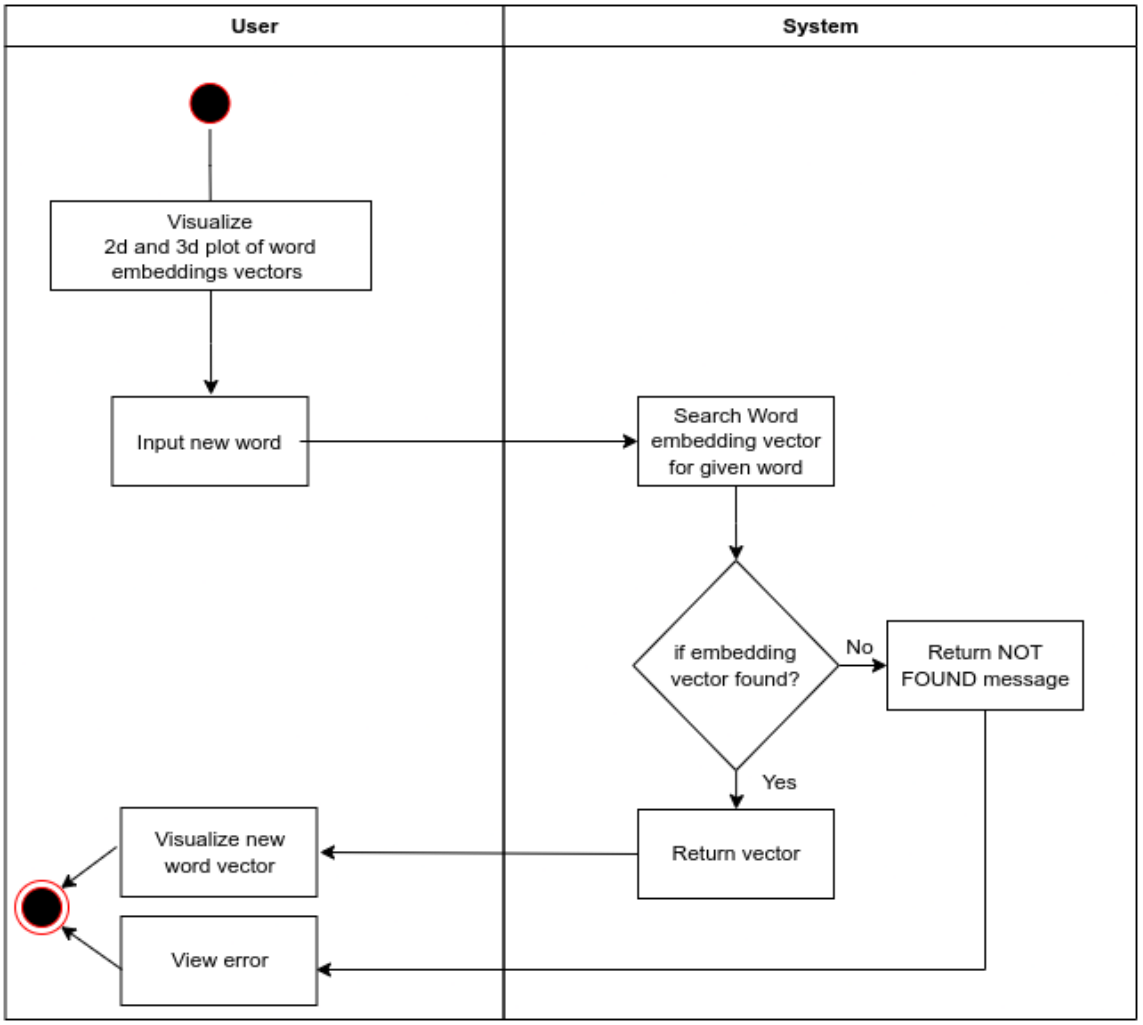
Sentimental Classification activity diagram



Spelling Correction Activity Diagram

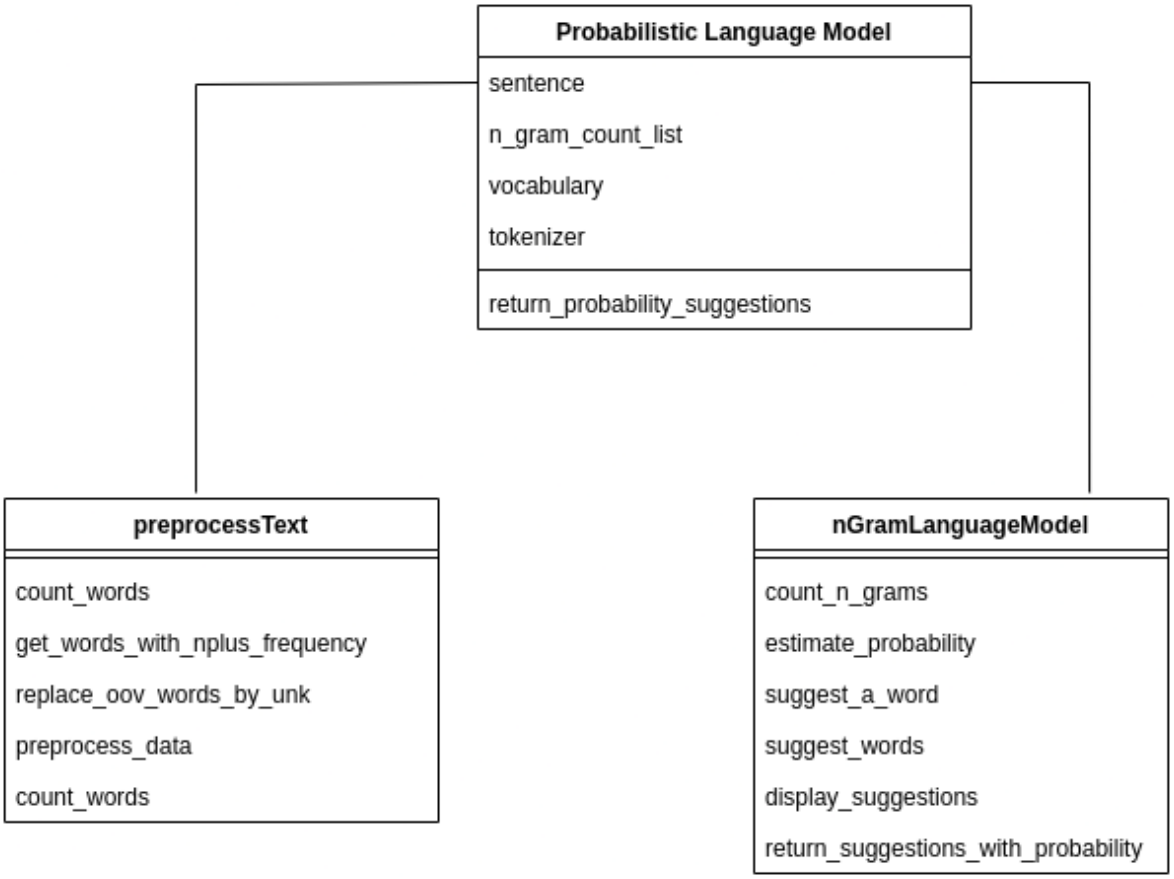


Word Embedding Activity Diagram

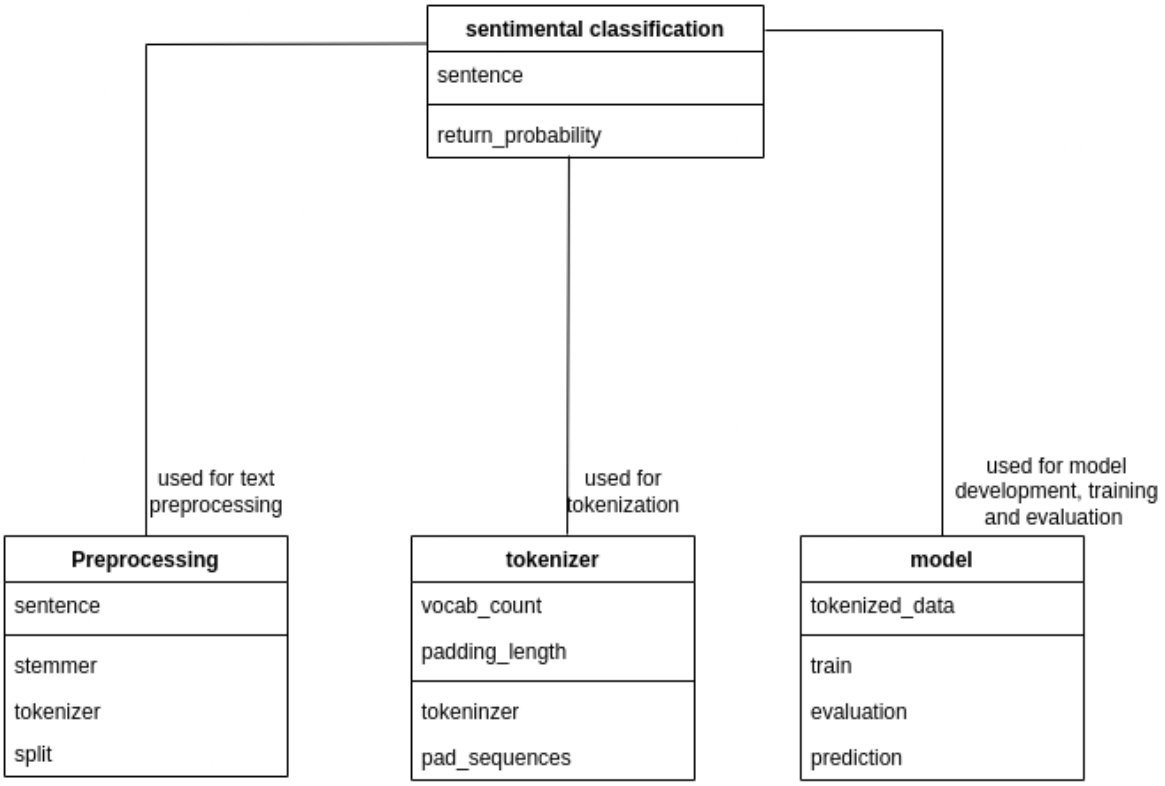


UML Class Diagram

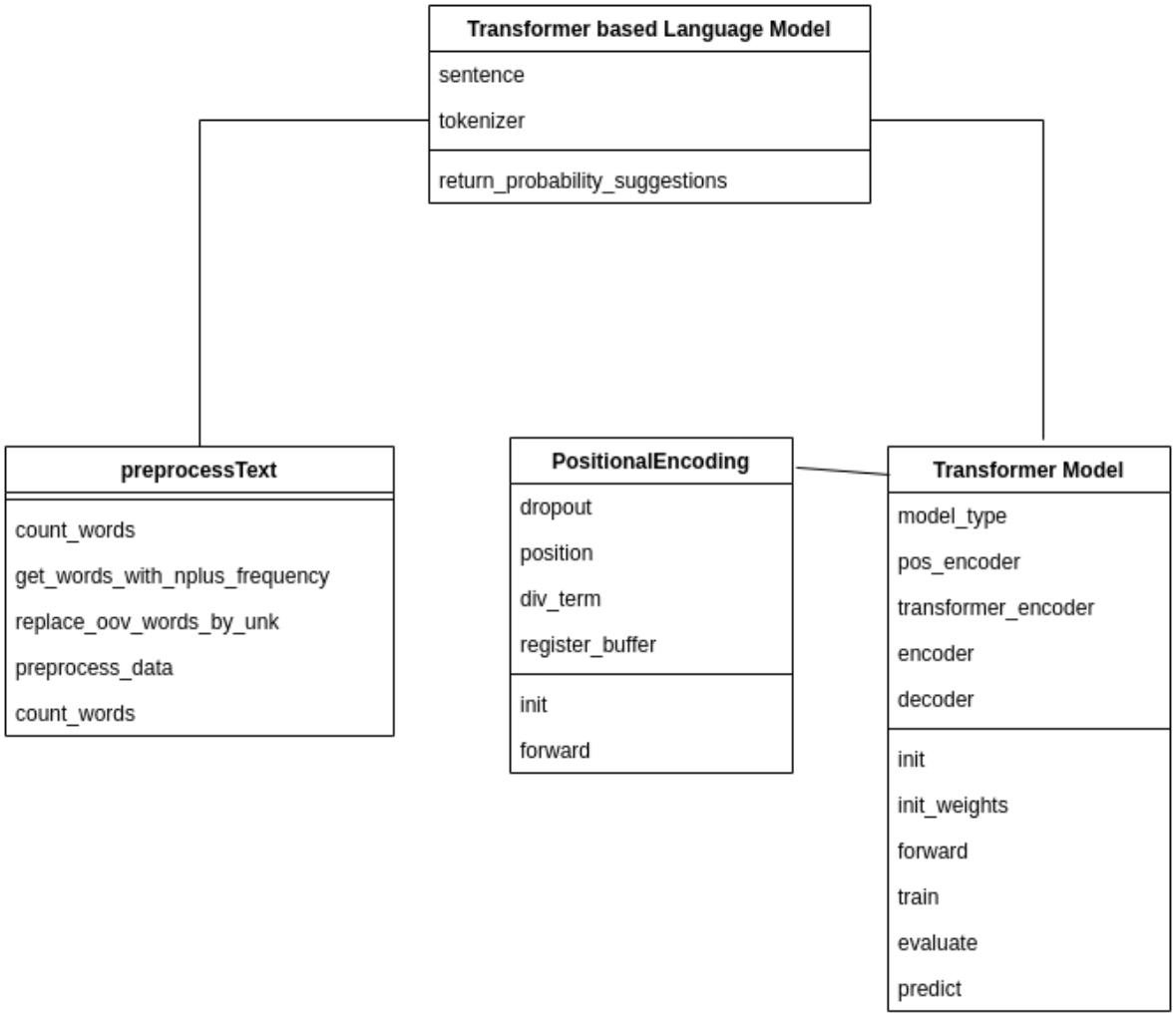
Probabilistic Language model class diagram



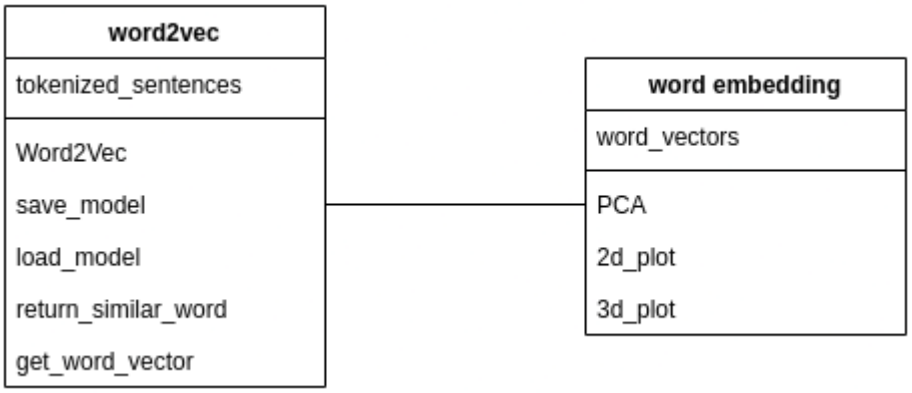
Sentimental Classification class diagram



Transformer based LM class diagram



Word Embedding class diagram



UML Sequence Diagram