**all\_words =** ['%', ',', '...', '60', 'about', 'achiev', 'aid', 'anyon', 'appli', 'applic', 'can', 'criteria', 'doe', 'financi', 'for', 'fsc', 'get', 'guid', 'have', 'hazara', 'how', 'i', 'in', 'inform', 'is', 'less', 'me', 'need', 'of', 'offer', 'piea', 'procedur', 'scholarship', 'student', 'than', 'the', 'those', 'to', 'univers', 'what', 'who']

**tags =** ['admission', 'scholarship']

**xy =** [(['how', 'can', 'i', 'get', 'scholarship', 'in', 'hazara', 'university', '?'], 'scholarship'), (['does', 'hazara', 'university', 'offer', 'scholarship', 'to', 'students', '?'], 'scholarship'), (['criteria', 'for', 'scholarship'], 'scholarship'), (['scholarship'], 'scholarship'), (['financial', 'aid'], 'scholarship'), (['Can', 'anyone', 'guide', 'me', 'about', 'scholarship', '...', '?'], 'scholarship'), (['I', 'need', 'information', 'about', 'the', 'scholarship', 'procedure', '.'], 'scholarship'), (['How', 'can', 'I', 'apply', 'in', 'Hazara', 'University', '?'], 'admission'), (['I', 'have', 'less', 'than', '60', '%', 'in', 'FSC', ',', 'How', 'can', 'I', 'apply', 'in', 'Hazara', 'University', '?'], 'admission'), (['What', 'is', 'the', 'procedure', 'of', 'application', 'in', 'Hazara', 'University', 'for', 'those', 'students', 'who', 'have', 'achieved', 'less', 'than', '60', '%', 'in', 'FSC', '?'], 'admission')]

**all\_words:** This is a list of all the different words that appear in the sentences provided in the data.

**tags:** These are the categories or labels that the sentences belong to. For example, if the sentences are about greetings, the tag might be "greeting".

**xy:** list of tuple containing every pattern and its tag.

**Bag of Words (BoW) Example:**

Let's say we have a simple dataset with three sentences:

**"I love machine learning"**

**"Machine learning is fascinating"**

**"I enjoy learning"**

To represent these sentences using the bag-of-words approach, we first need to create a vocabulary, which consists of all unique words in the dataset. In this case, the vocabulary would be:

**['I', 'love', 'machine', 'learning', 'is', 'fascinating', 'enjoy']**

Now, we represent each sentence as a vector indicating the presence or absence of each word in the vocabulary. For example:

**Sentence 1: [1, 1, 1, 1, 0, 0, 0] (indicating presence of "I", "love", "machine", "learning")**

**Sentence 2: [0, 0, 1, 1, 1, 1, 0] (indicating presence of "machine", "learning", "is", "fascinating")**

**Sentence 3: [1, 0, 0, 1, 0, 0, 1] (indicating presence of "I", "learning", "enjoy")**

**Xtrain is bag of words for each sentence / pattern:**

[array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0.], dtype=float32), array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0., 1., 1., 0., 0.], dtype=float32), array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32), array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32), array([0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32), array([0., 0., 1., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32), array([0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0., 0.], dtype=float32), array([0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.], dtype=float32), array([1., 1., 0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 1., 1., 0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0.], dtype=float32), array([1., 0., 0., 1., 0., 1., 0., 0., 0., 1., 0., 0., 0., 0., 1., 1., 0., 0., 1., 1., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0., 1., 0., 1., 1., 1., 1., 0., 1., 1., 1.], dtype=float32)]

**Y\_train is :**y\_train is a list containing labels or intents corresponding to each input sentence in X\_train.

Each element in y\_train represents the label or intent of the corresponding input sentence.

The order of elements in y\_train corresponds to the order of input sentences in X\_train.

In your provided example, the labels are encoded as 0 for the "scholarship" intent and 1 for the "admission" intent.

E,g

X\_train = [

[0, 0, 0, 1, 1, 0, 0, ...], # Bag-of-words vector for sentence 1

[0, 0, 1, 1, 0, 0, 1, ...], # Bag-of-words vector for sentence 2

...

]

y\_train = [0, 0, 0, 1, 1, ...] # Labels (0 for "scholarship", 1 for "admission")