|  |  |
| --- | --- |
| **Information** | **Meaning** |
| Phonetic | How a word is expressed vocally with a certain sound. |
| Structural | How a word is composed of different linguistic components. |
| Syntactic | How a word fits into the overall structure of a sentence |
| Semantic | What is the meaning of a word in some particular context? |
| Pragmatic | How a word is used in a discourse or conversation. |

|  |  |  |
| --- | --- | --- |
| **English** | **Hindi** | **Tamil** |
| I will go | मैं जाऊंगा | நான் ேபாேவன |
| We will go | हम जायेंगे | நாம் ேபாேவாம |
| You will go | तुम जाओगे | நீ ேபாவாய |
| He will go | वह जाएगा | அவன் ேபாவான |
| She will go | वो जाएगी | அவள் ேபாவாள |

|  |  |  |
| --- | --- | --- |
| **Original Word** | **Stemming** | **Lemmatisation** |
| witnessed | wit | witness |
| assignments | assign | assignment |
| considerable | consider | considerable |
| democratisation | democrat | democratisation |
| interpolated | interpol | interpolate |
| effectively | effect | effective |

|  |  |  |
| --- | --- | --- |
| **Algorithm 1** The steps for subword tokenisation as adopted by BPE and WordPiece. The actual formula in Step 4 is realised by Equations (2.1) and (2.2), respectively, for BPE and WordPiece. |  | **Algorithm 2** Algorithm for obtaining unique tokens in the corpus via splitting at the word level. |
| **Input:** Vocabulary size *k*, Corpus *D*, Maximum Iteration *maxiter*  **Output:** Vocabulary V  1: *V* ***←*** *PREPROCESS(D)*  2: *i* ***←*** 0  3: **while |***V****|*** *< k* or *i* <*maxiter* **do**  4: *tl* : *tr* ***←*** *max (FC (t l, t r))*  5: *tlr* ***←*** *tl : tr*  6: *tl*:*tr* ***←*** *tlr*  7: *V* ***←*** *V* **∩** *tlr*  8: *i* ***←*** *i* + 1  9: **end while**  **return** V |  | **Input:** Corpus *D*  **Output:** Vocabulary V  1: *V* ***←***{}  2: **for** *w* ∈*split(D, delimiter=" ")* **do**  3: **for** *ch* ∈*w* **do**  4: *V* **∩** *ch*  5: **end for**  6: **end for**  **return** V |

|  |  |  |
| --- | --- | --- |
| *x*1 | *x*2 | *x*1 AND *x*2 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| *x*1 | *x*2 | *x*1 OR *x*2 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| *x*1 | *x*2 | *x*1 XOR *x*2 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x*1 | *x*2 | *x*1 + *x*2 – 1*.*5 | *y* = *sgn'*(*x*1+ *x*2***–*** 1*.*5) | *x*1 AND *x*2 |
| 0 | 0 | –1.5 | **0** | **0** |
| 0 | 1 | –0.5 | **0** | **0** |
| 1 | 0 | –0.5 | **0** | **0** |
| 1 | 1 | 0.5 | **1** | **1** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Units** | **Hidden Units** | | **Output** | ***x*1 XOR *x*2** |
| ***x*1** | ***x*2** | ***h*1 *=* sgn'(*x*1+ *x*2 *–* 0.5)** | ***h*2 *=* sgn'(*x*1+ *x*2 *–* 1*.*5)** | ***y =* sgn'(*h*1 *– h*2 *–* 0*.*5)** |
| 0  0  1  1 | 0  1  0  1 | sgn*'*(0 · 1 + 0 · 1 – 0*.*5) = 0  sgn*'*(0 · 1 + 1 · 1 – 0*.*5) = 1  sgn*'*(1 · 1 + 0 · 1 – 0*.*5) = 1  sgn'(1 · 1 + 1 · 1 – 0*.*5) = 1 | sgn'(0 · 1 + 0 · 1 – 1*.*5) = 0  sgn'(0 · 1 + 1 · 1 – 1*.*5) = 0  sgn'(1 · 1 + 0 · 1 – 1*.*5) = 0  sgn'(1 · 1 + 1 · 1 – 1*.*5) = 1 | sgn*'*(0 · 1 + 0 · – 1 – 0*.*5) = **0**  sgn*'*(1 · 1 + 0 · – 1 – 0*.*5) = **1**  sgn*'*(1 · 1 + 0 · – 1 – 0*.*5) = **1**  sgn*'*(1 · 1 + 1 · – 1 – 0*.*5) = **0** | **0**  **1**  **1**  **0** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Expected** *y* | 1 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 |
| **Predicted** *ŷ* | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 | 1 | -1 |
| **Type** | TP | FN | FP | TP | TN | FP | TP | TP | TP | FN |

|  |  |  |
| --- | --- | --- |
| Actual | Predicted | |
| Positive | Negative |
| Positive | 5 (TP) | 2 (FN) |
| Negative | 2 (FP) | 1 (TN) |