SignScorerProduct

Utility classes for combining scorers multiplicatively or via linear interpolation.

RepetitionScorer
+penalty: double
+updateContext(sign: Sign)
+resetContext( )
+ageContext( )

Returns a score that is linear in log space with the number of repeated items times the penalty.

«interface»
SignScorer

+score(sign: Sign, complete: boolean): double
+nullScorer: SignScorer

SignScorerInterpolation
#weights: double[ ]

Returns zero for all signs, thereby providing no distinguishing information.

Utility class for interpolating n–gram models at the word level.

LinearNgramScorerCombo
#weights: double[ ]
#logProbFromNgram(i: int, order: int): float

NgramScorer
#order: int
#reverse: boolean
#useSemClasses: boolean
#wordsToScore: List<Word>
#prepareToScoreWords( )
#score( ): double
#logProbFromNgram(i: int, order: int): float

«interface»
NgramFilter

+filterOut(words: List<Word>): boolean

Filters bigrams with the wrong choice of a or an given the initial letter of the following word, with configurable exceptions.

AAnFilter
+addException(w1: String, w2: String)

N–gram models can be of any order, can reverse the words, and can replace certain words with their semantic classes.

StandardNgramModel
+StandardNgramModel(order: int, filename:String)
#prepareToScoreWords( )
#logProbFromNgram(i: int, order: int): float

NgramPrecisionModel
+NgramPrecisionModel(targets: String[ ], order: int)
#prepareToScoreWords( )
#score( ): double

Returns a modified version of the BLEU score used in MT evaluation.

FactoredNgramModel
+FactoredNgramModel(child: String, parents: String[ ], filename: String)
#prepareToScoreWords( )
#logProbFromNgram(i: int, order: int): float

FactoredNgramModelFamily
+FactoredNgramModelFamily(filename: String)
#prepareToScoreWords( )
#logProbFromNgram(i: int, order: int): float

Factored n–gram models return the probability of the child factor of the current word given a sequence of parent factors. Multiple models can be organized into families.