universität
uulm

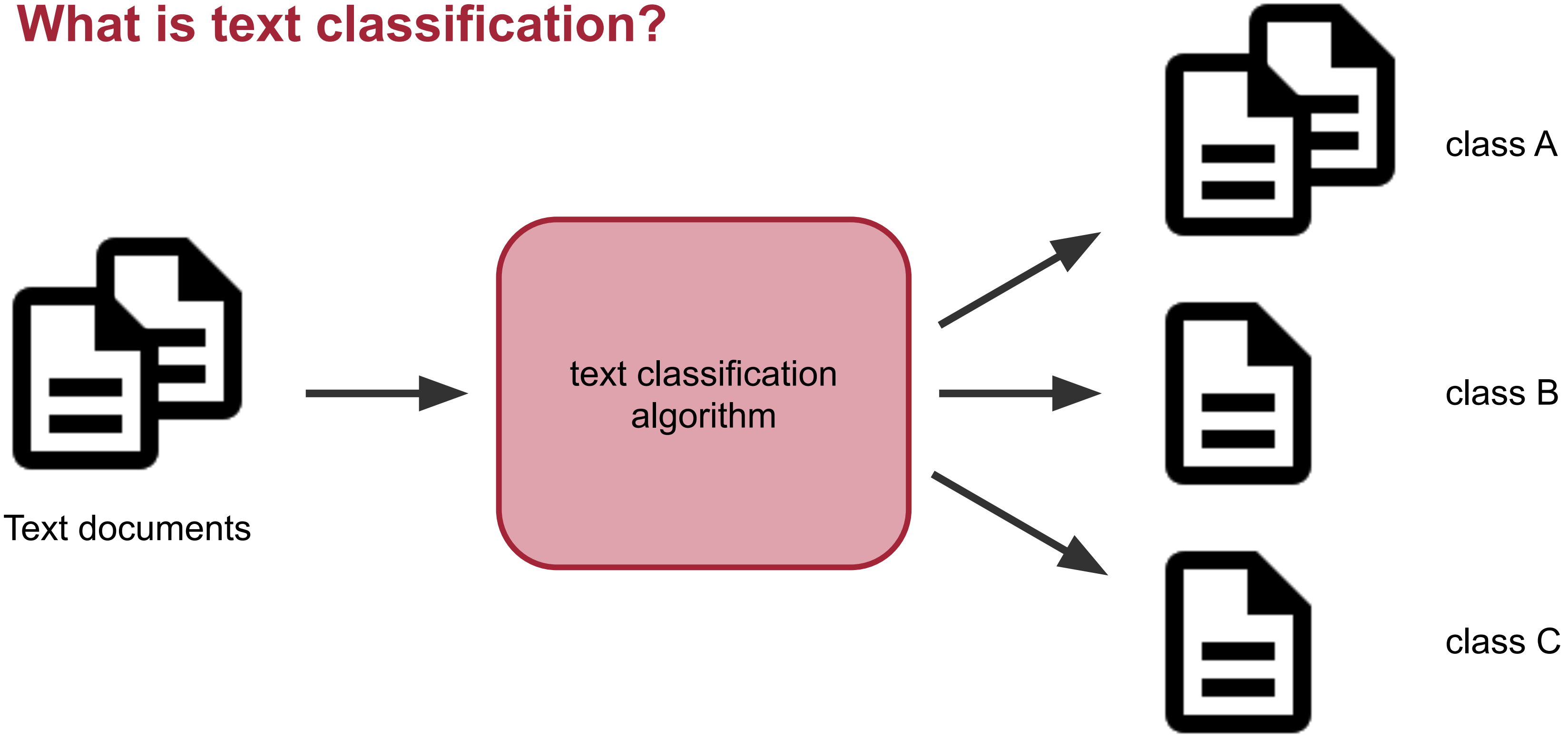
Lighweight Text Classification

Sarah Marek | 07.03.2022

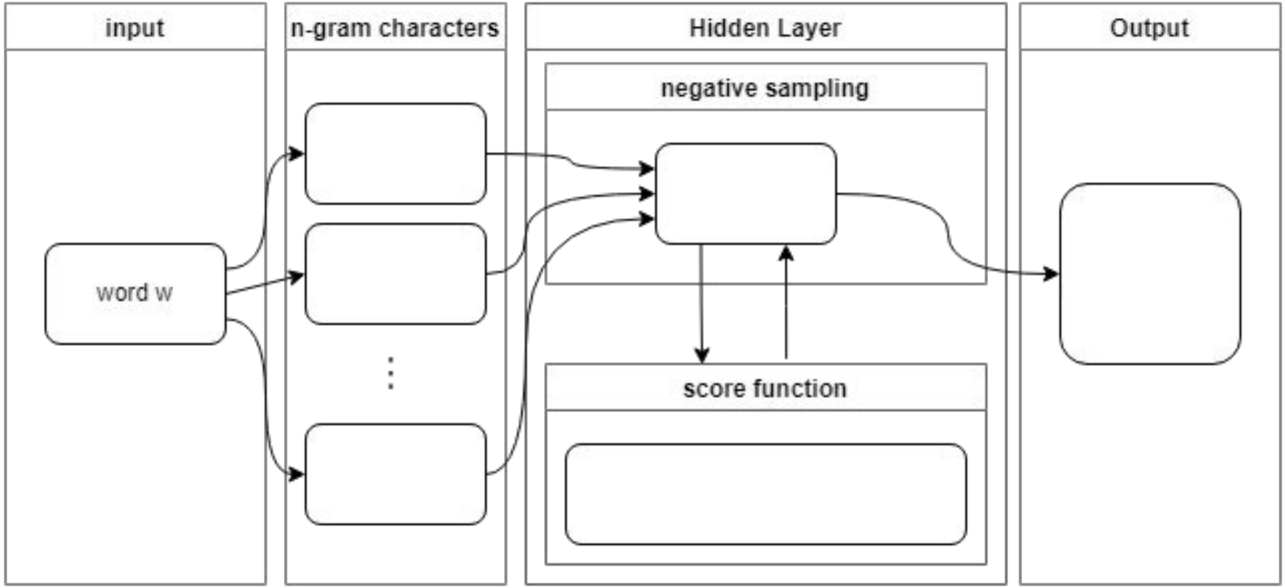
Table of contents

- What is text classification?
- skip-gram
- FastText
- MDLText
- Results
- Conclusion

What is text classification?

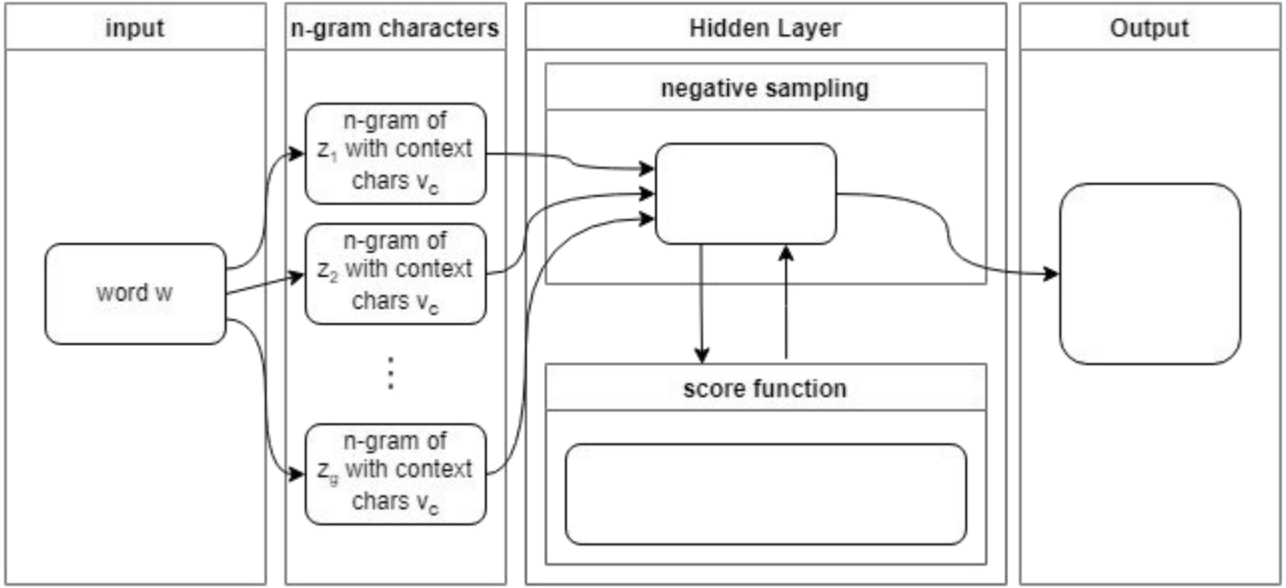


skip-gram


input
n-gram characters
Hidden Layer
Output
negative sampling
word w
⋮
score function

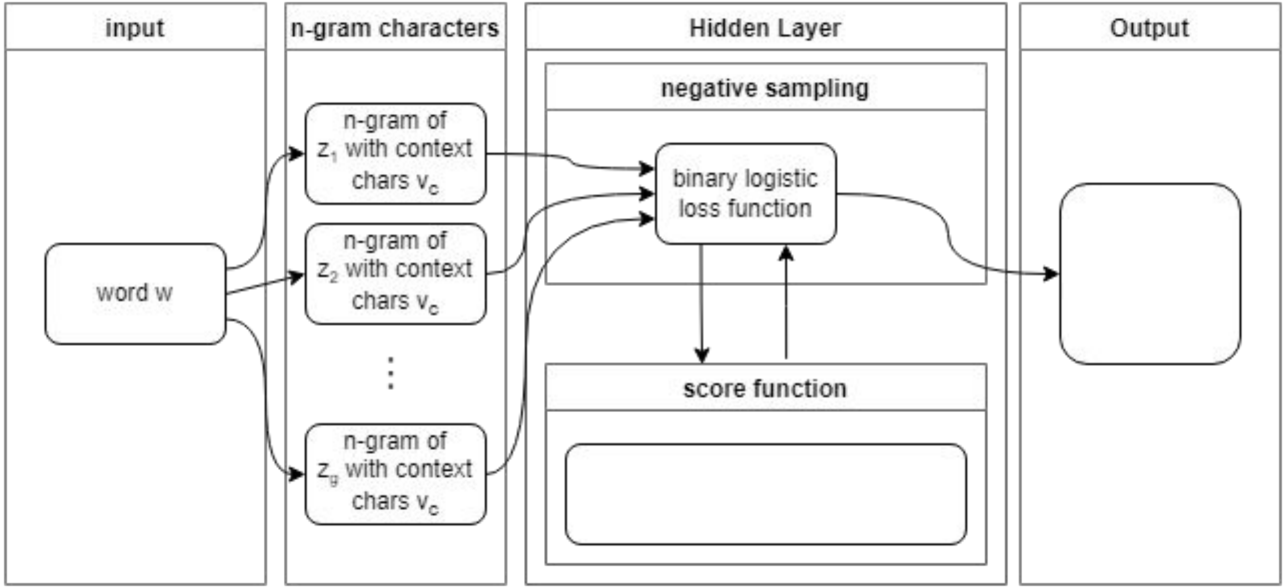
skip-gram

n-grams: house $\xrightarrow{\text{2-grams}}$ {ho, ou, us, se}

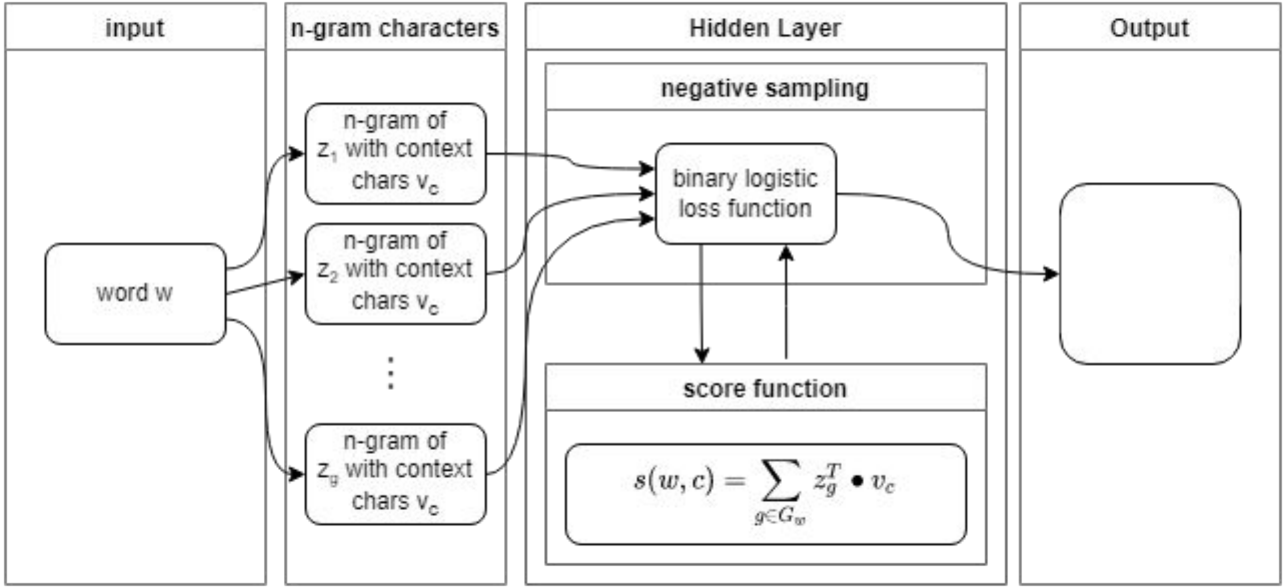

input
n-gram characters
Hidden Layer
Output
word w
n-gram of z_1 with context chars v_c
n-gram of z_2 with context chars v_c
⋮
n-gram of z_g with context chars v_c
negative sampling
score function

skip-gram

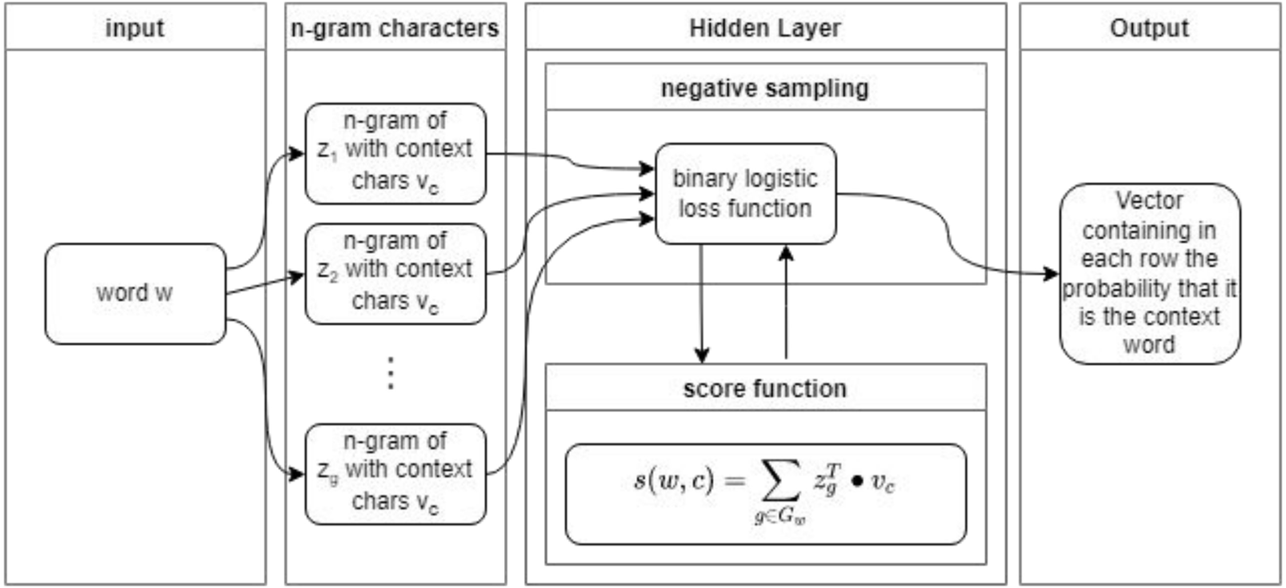
binary logistic loss function: $\sum_{t=1}^{T}\left[\sum_{c\in C_t} l(s(w_t,w_c)) + \sum_{n\in N_{t,c}} l(-s(w_t,n))\right]; \; l \rightarrow \log(1+e^{-x})$


input
n-gram characters
Hidden Layer
Output
word w
n-gram of z_1 with context chars v_c
n-gram of z_2 with context chars v_c
⋮
n-gram of z_g with context chars v_c
negative sampling
binary logistic loss function
score function

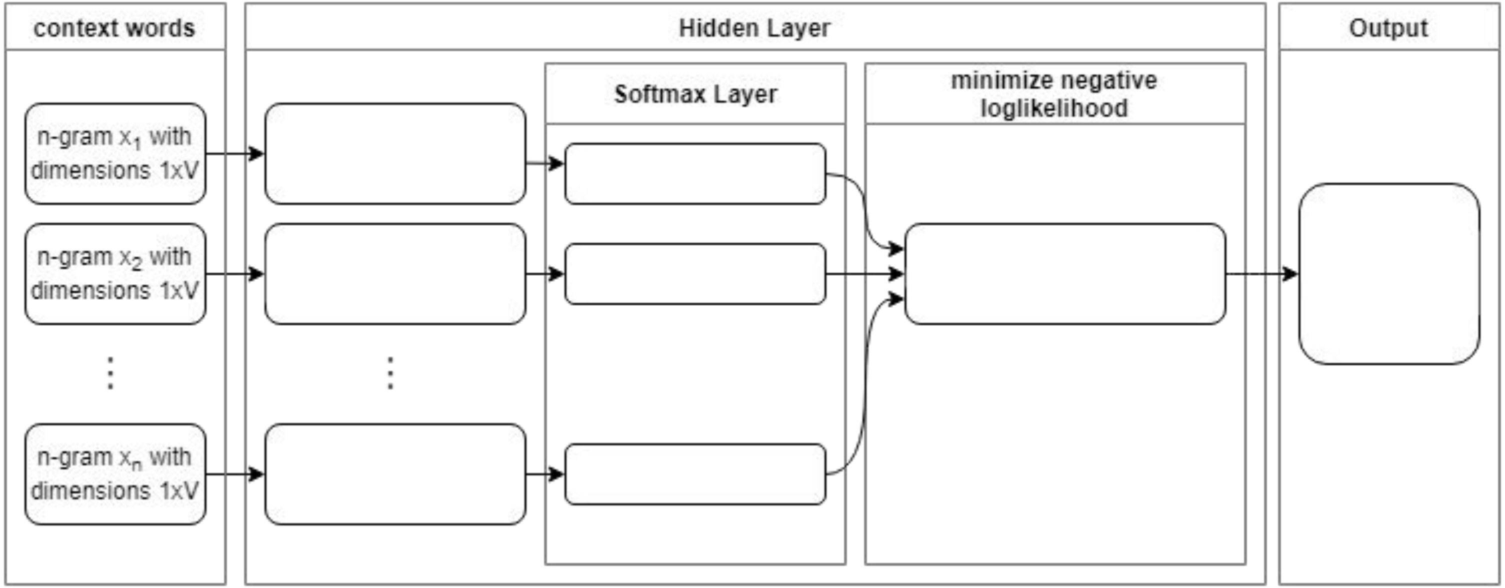
skip-gram


input
n-gram characters
Hidden Layer
Output
word w
n-gram of z_1 with context chars v_c
n-gram of z_2 with context chars v_c
⋮
n-gram of z_g with context chars v_c
negative sampling
binary logistic loss function
score function
$s(w,c) = \sum_{g \in G_w} z_g^T \bullet v_c$

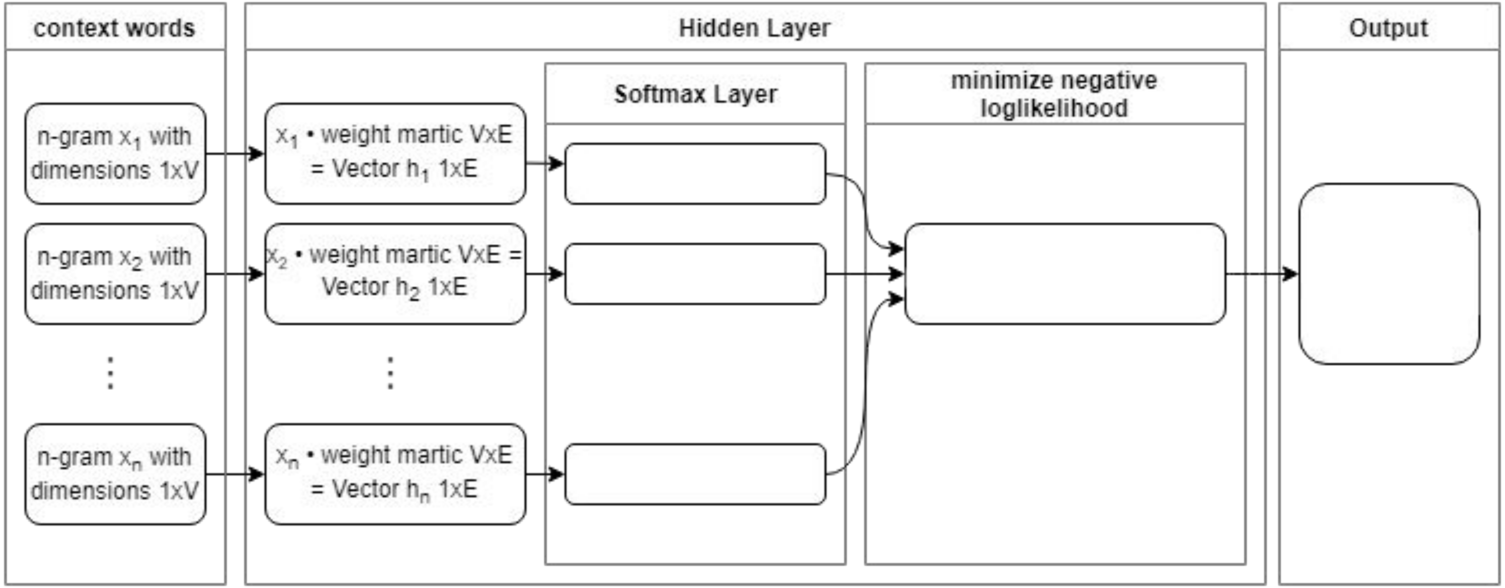
skip-gram


input
n-gram characters
Hidden Layer
Output
word w
n-gram of z_1 with context chars v_c
n-gram of z_2 with context chars v_c
⋮
n-gram of z_g with context chars v_c
negative sampling
binary logistic loss function
score function
$s(w,c) = \sum_{g \in G_w} z_g^T \bullet v_c$
Vector containing in each row the probability that it is the context word

fastText

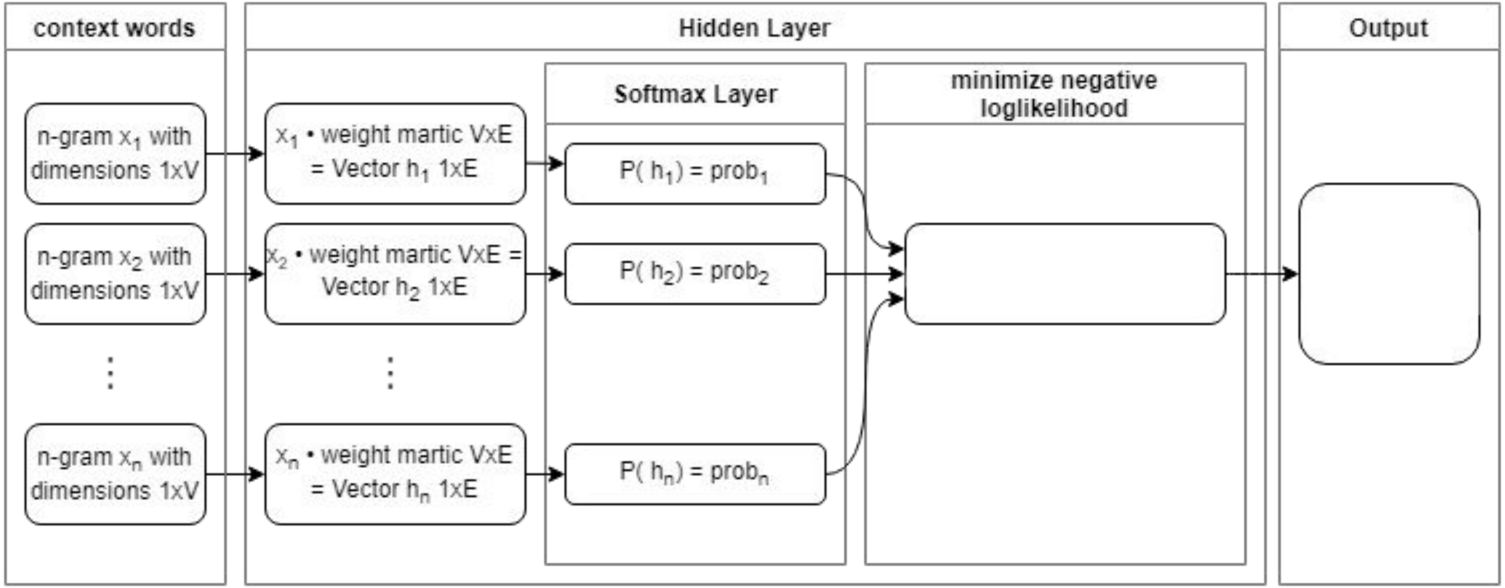

context words
Hidden Layer
Output
Softmax Layer
minimize negative loglikelihood
n-gram x_1 with dimensions 1xV
n-gram x_2 with dimensions 1xV
n-gram x_n with dimensions 1xV

fastText

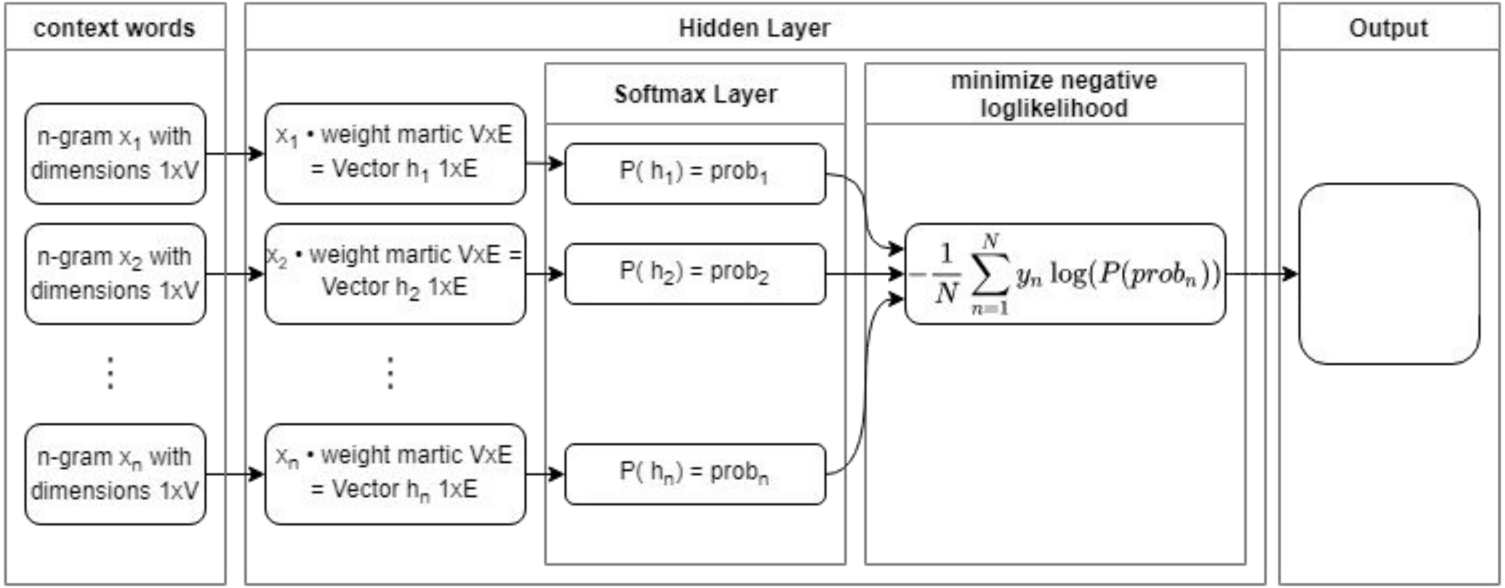

context words
Hidden Layer
Output
Softmax Layer
minimize negative loglikelihood
n-gram x_1 with dimensions 1xV
n-gram x_2 with dimensions 1xV
⋮
n-gram x_n with dimensions 1xV
x_1 • weight martic VxE = Vector h_1 1xE
x_2 • weight martic VxE = Vector h_2 1xE
⋮
x_n • weight martic VxE = Vector h_n 1xE

fastText

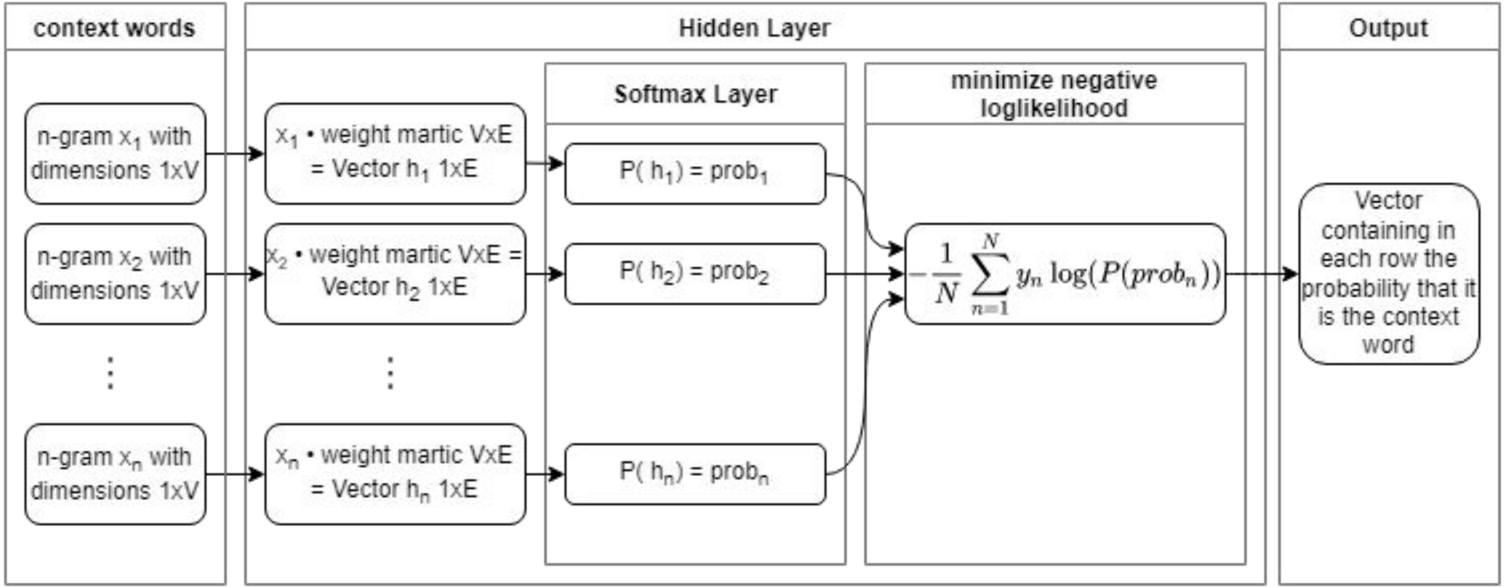
Softmax: $P(w_1 w_n) = \prod_{i=1}^{n} P(w_i|w_1, w_{i-1})$; $P(w|w_1 w_{i-1}) = \frac{exp(\sum_j \lambda_j f_j(w, w1w_{i-1}))}{Z_\lambda(w_1 w_{i-1})}$


context words
Hidden Layer
Output
n-gram x_1 with dimensions 1xV
n-gram x_2 with dimensions 1xV
n-gram x_n with dimensions 1xV
x_1 • weight martic VxE = Vector h_1 1xE
x_2 • weight martic VxE = Vector h_2 1xE
x_n • weight martic VxE = Vector h_n 1xE
Softmax Layer
P(h_1) = $prob_1$
P(h_2) = $prob_2$
P(h_n) = $prob_n$
minimize negative loglikelihood

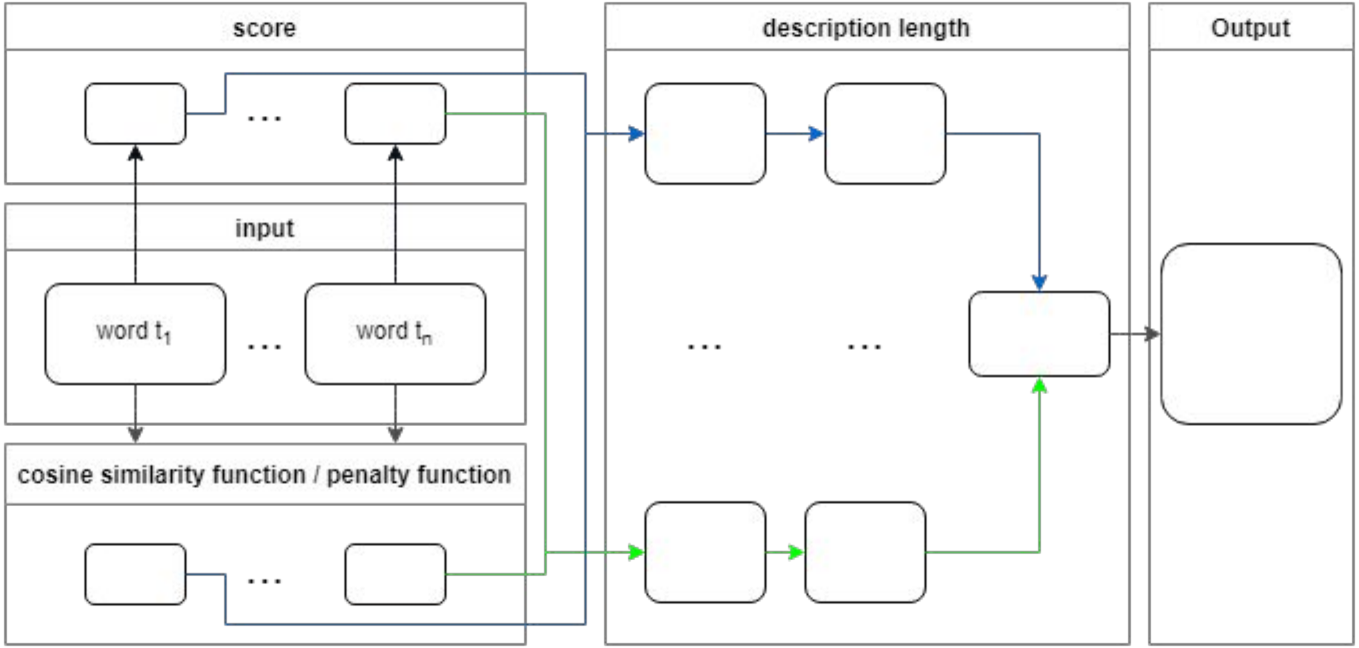
fastText


context words
Hidden Layer
Output
n-gram x_1 with dimensions 1xV
n-gram x_2 with dimensions 1xV
⋮
n-gram x_n with dimensions 1xV
x_1 • weight martic VxE = Vector h_1 1xE
x_2 • weight martic VxE = Vector h_2 1xE
⋮
x_n • weight martic VxE = Vector h_n 1xE
Softmax Layer
P(h_1) = $prob_1$
P(h_2) = $prob_2$
P(h_n) = $prob_n$
minimize negative loglikelihood
$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(P(prob_n))$

fastText

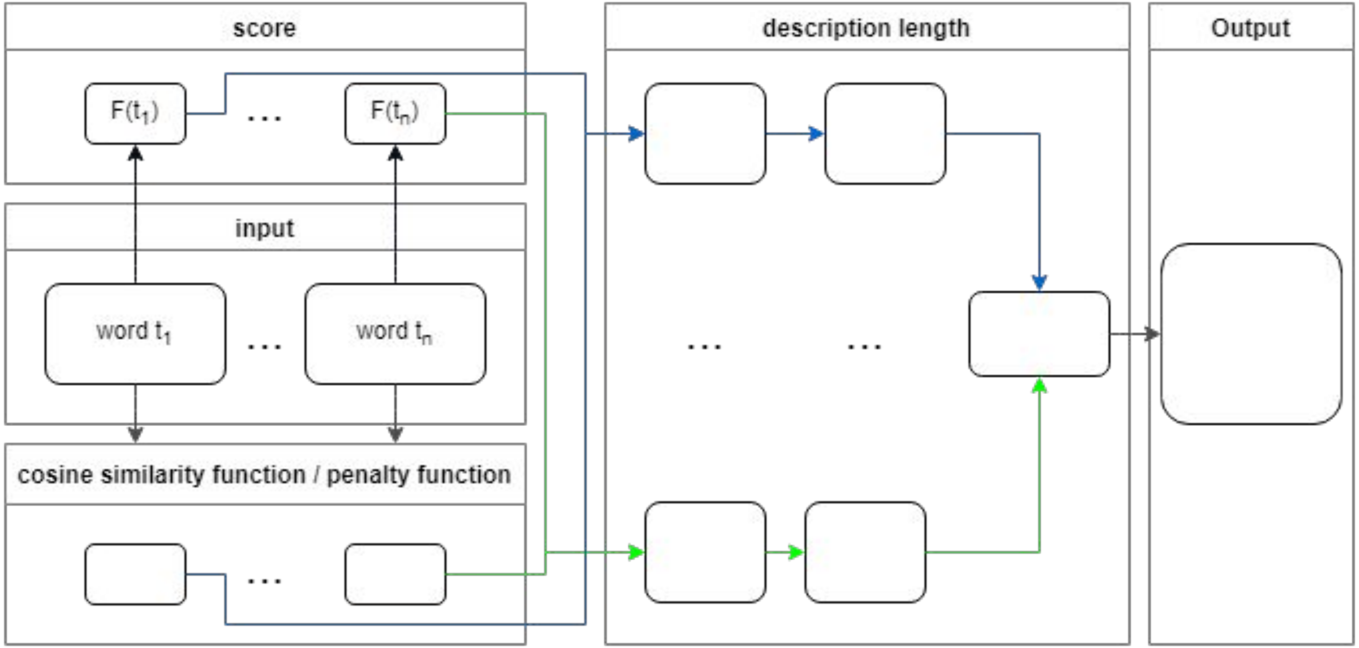

context words
Hidden Layer
Output
n-gram x_1 with dimensions 1xV
n-gram x_2 with dimensions 1xV
⋮
n-gram x_n with dimensions 1xV
x_1 • weight martic VxE = Vector h_1 1xE
x_2 • weight martic VxE = Vector h_2 1xE
⋮
x_n • weight martic VxE = Vector h_n 1xE
Softmax Layer
P(h_1) = $prob_1$
P(h_2) = $prob_2$
P(h_n) = $prob_n$
minimize negative loglikelihood
$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(P(prob_n))$
Vector containing in each row the probability that it is the context word

MDLText

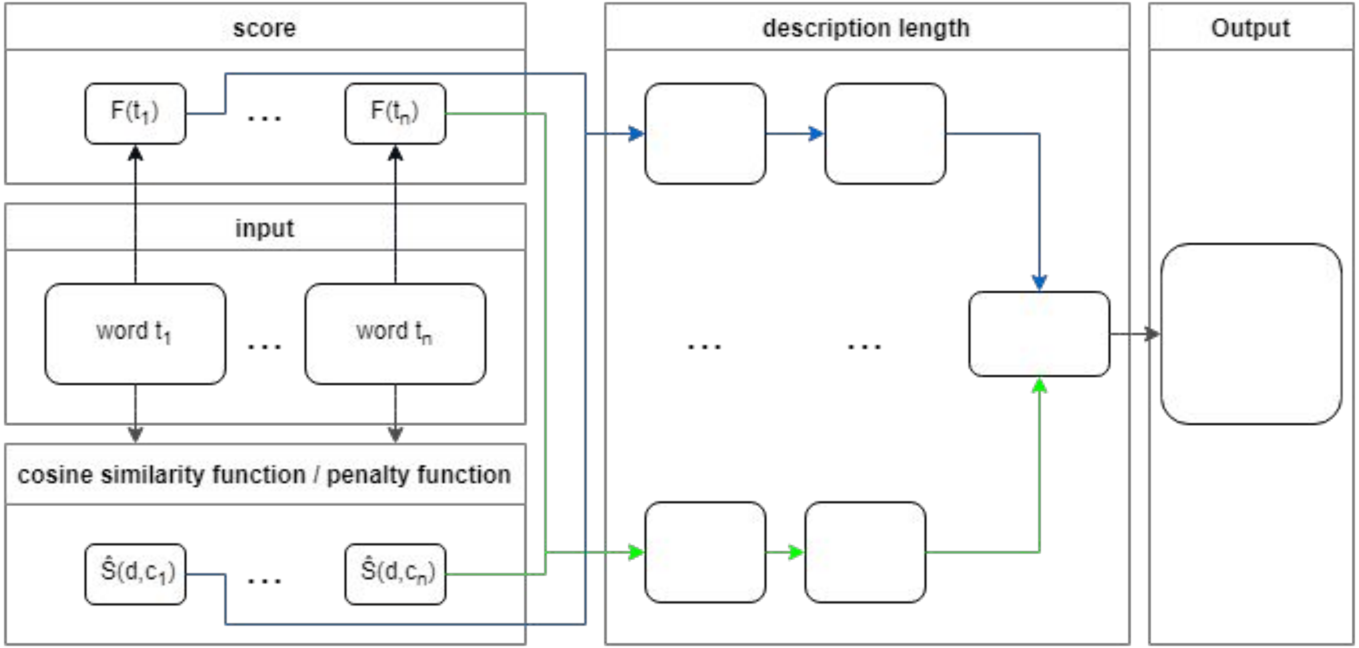

score
input
word t_1
...
word t_n
cosine similarity function / penalty function
description length
Output

MDLText

$$0 \leq F(t_i) \leq 1; \; \hat{S}(d, c_j) = -\log_2(\frac{1}{2} \times S(d, \bar{c}))$$

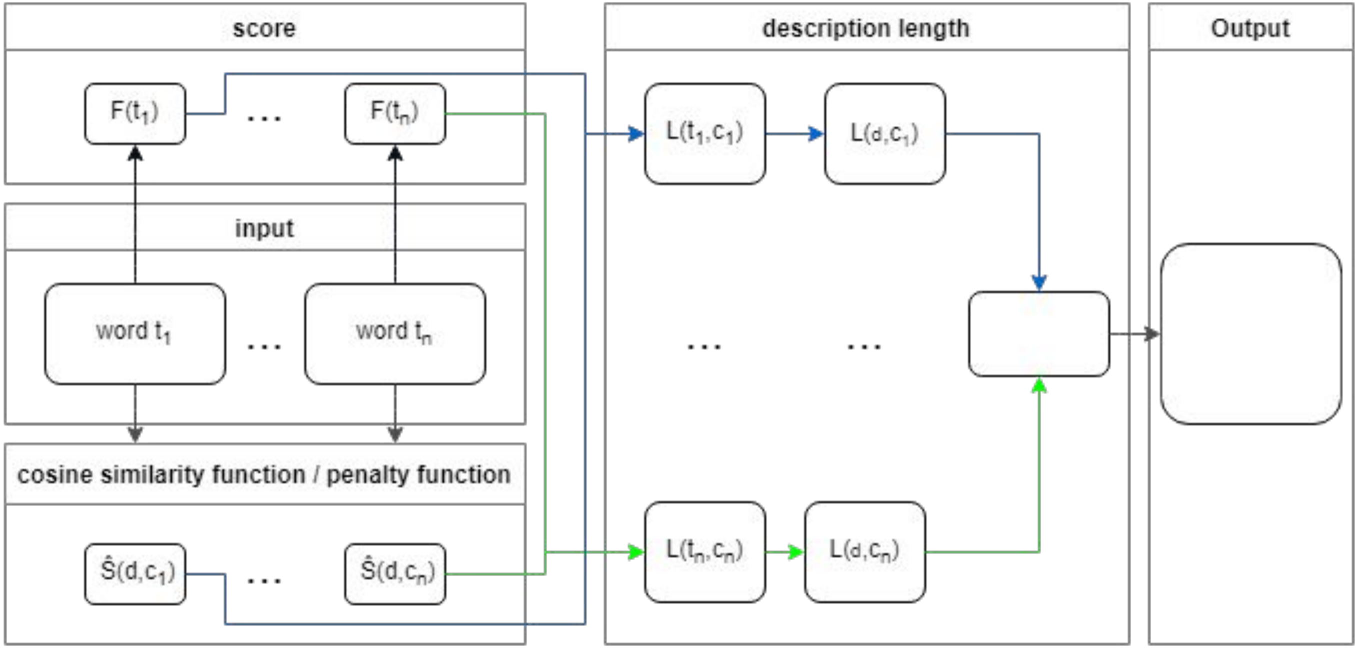


MDLText


score
$F(t_1)$
. . .
$F(t_n)$
input
word t_1
. . .
word t_n
cosine similarity function / penalty function
$\hat{S}(d,c_1)$
. . .
$\hat{S}(d,c_n)$
description length
. . .
. . .
Output

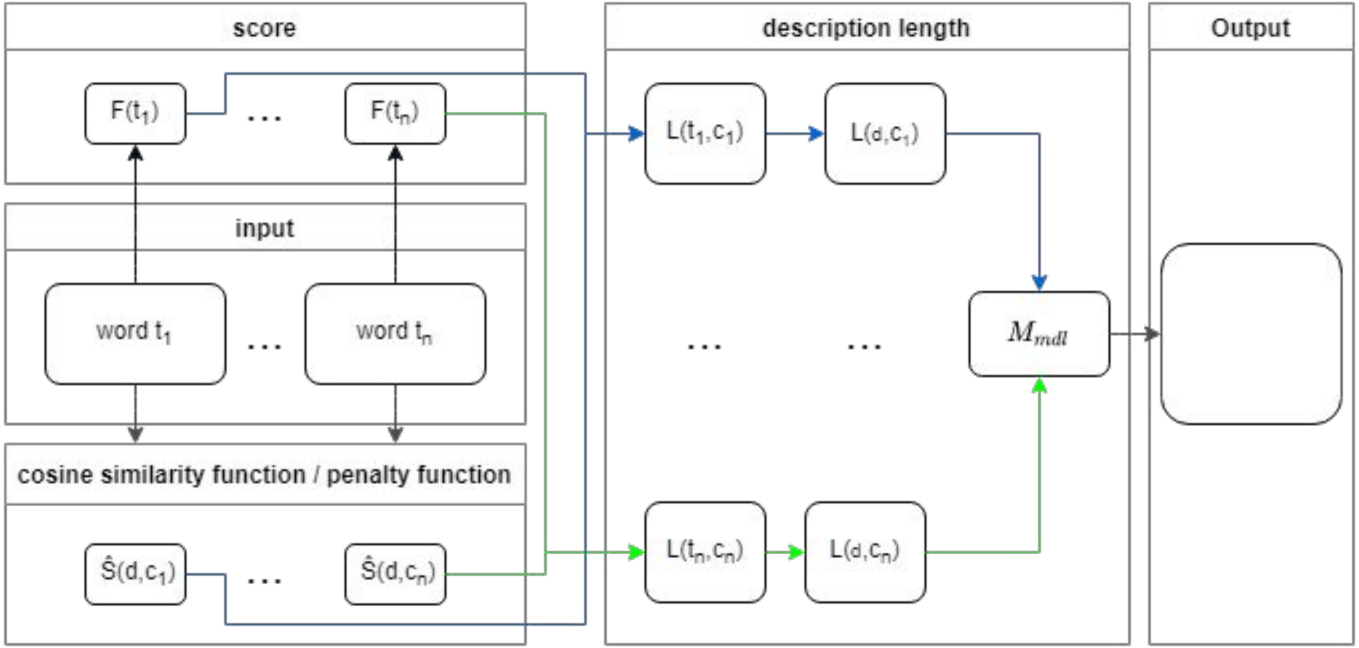
MDLText

$$L(t_i|c_j) = \lceil -\log_2 \beta(t_i|c_j) \rceil ; L(d|c_j) = L(d|c_j) + (L(t_i, c_j) \ \times \ F(t_i))$$

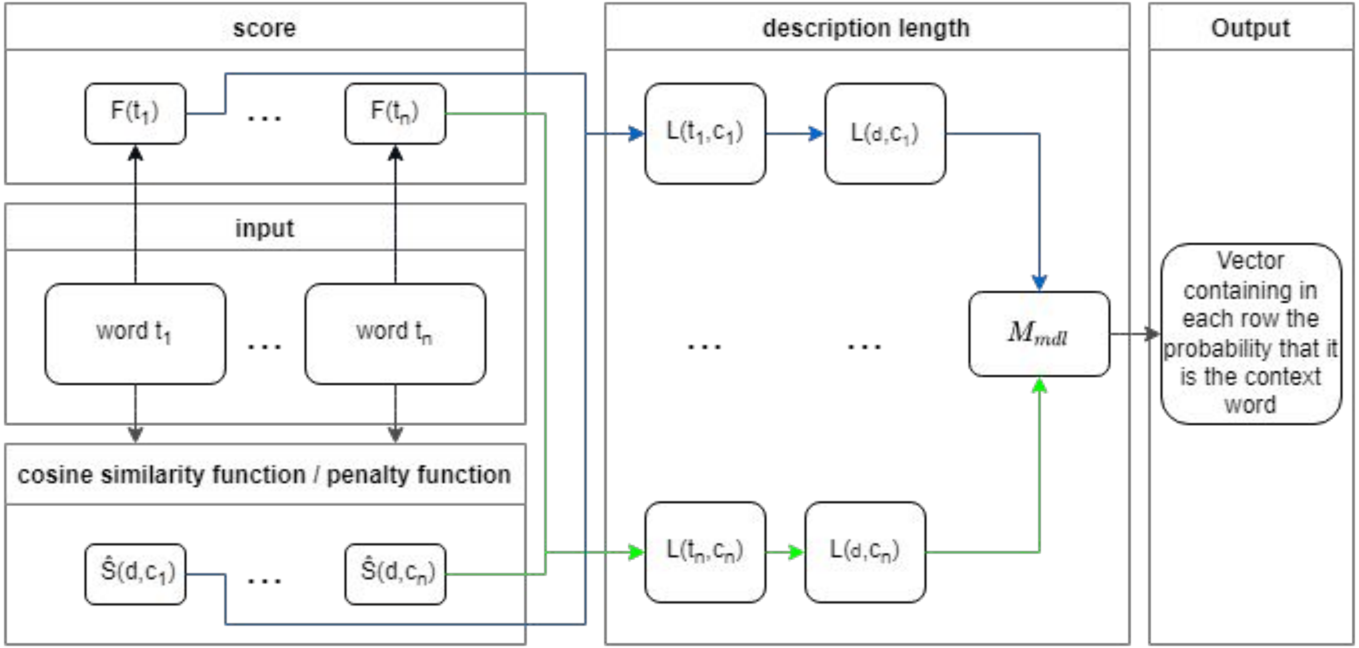


MDLText

$$M_{mdl} = \arg\min_{\forall M} L(d|c_j)$$



MDLText

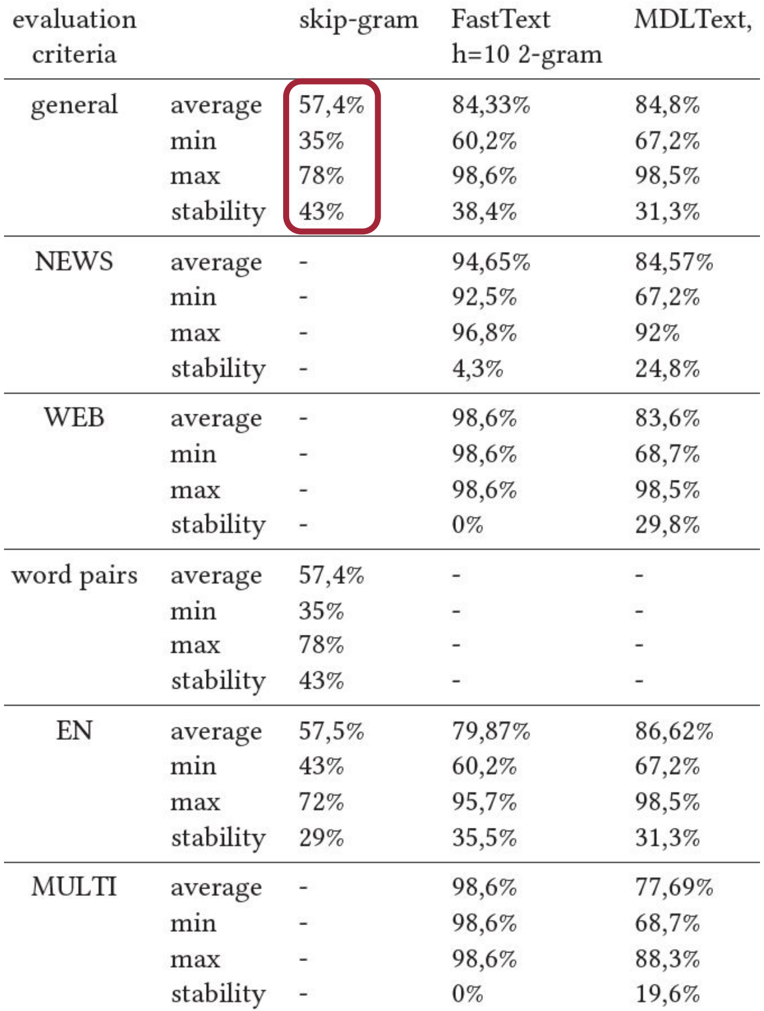

score
$F(t_1)$
...
$F(t_n)$
input
word t_1
...
word t_n
cosine similarity function / penalty function
$\hat{S}(d,c_1)$
...
$\hat{S}(d,c_n)$
description length
$L(t_1,c_1)$
$L(d,c_1)$
...
...
M_{mdl}
$L(t_n,c_n)$
$L(d,c_n)$
Output
Vector containing in each row the probability that it is the context word

Results skip-gram

- lower average accuracy than the others
- word pair input
  - no context of the word
- stability is poor with 43%

Application area:

- common words as input
  - specific application
- 1-word subword recognition
  - search engine



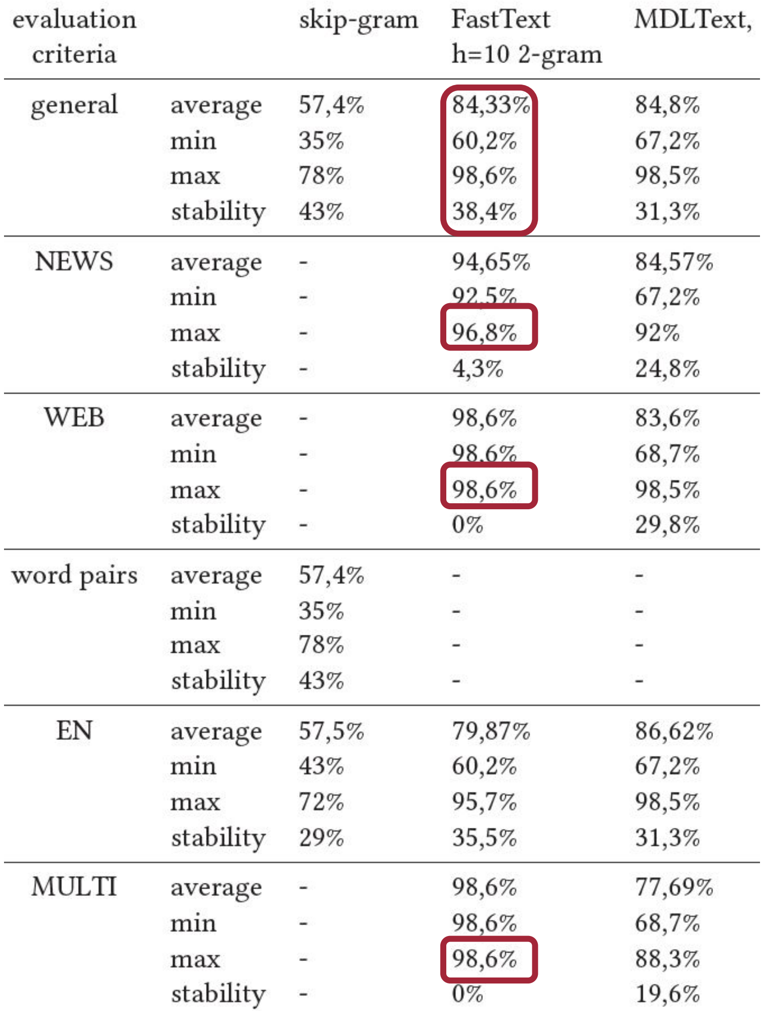
evaluation criteria		skip-gram	FastText h=10 2-gram	MDLText,
general	average	57,4%	84,33%	84,8%
	min	35%	60,2%	67,2%
	max	78%	98,6%	98,5%
	stability	43%	38,4%	31,3%
NEWS	average	-	94,65%	84,57%
	min	-	92,5%	67,2%
	max	-	96,8%	92%
	stability	-	4,3%	24,8%
WEB	average	-	98,6%	83,6%
	min	-	98,6%	68,7%
	max	-	98,6%	98,5%
	stability	-	0%	29,8%
word pairs	average	57,4%	-	-
	min	35%	-	-
	max	78%	-	-
	stability	43%	-	-
EN	average	57,5%	79,87%	86,62%
	min	43%	60,2%	67,2%
	max	72%	95,7%	98,5%
	stability	29%	35,5%	31,3%
MULTI	average	-	98,6%	77,69%
	min	-	98,6%	68,7%
	max	-	98,6%	88,3%
	stability	-	0%	19,6%

Results fastText

- second best
- best in some criterias
  - not enough test cases
  - still better than the best of MDLText in these cases

Application area:

- information through context words
  - well-written sentences



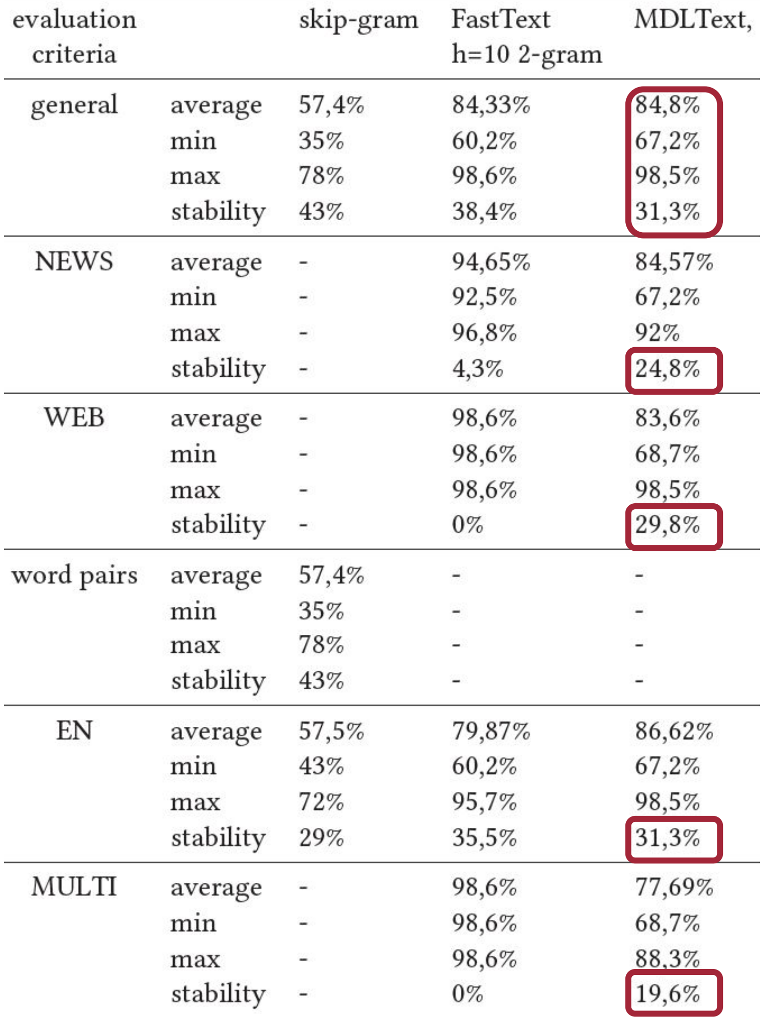
evaluation criteria		skip-gram	FastText h=10 2-gram	MDLText,
general	average	57,4%	84,33%	84,8%
	min	35%	60,2%	67,2%
	max	78%	98,6%	98,5%
	stability	43%	38,4%	31,3%
NEWS	average	-	94,65%	84,57%
	min	-	92,5%	67,2%
	max	-	96,8%	92%
	stability	-	4,3%	24,8%
WEB	average	-	98,6%	83,6%
	min	-	98,6%	68,7%
	max	-	98,6%	98,5%
	stability	-	0%	29,8%
word pairs	average	57,4%	-	-
	min	35%	-	-
	max	78%	-	-
	stability	43%	-	-
EN	average	57,5%	79,87%	86,62%
	min	43%	60,2%	67,2%
	max	72%	95,7%	98,5%
	stability	29%	35,5%	31,3%
MULTI	average	-	98,6%	77,69%
	min	-	98,6%	68,7%
	max	-	98,6%	88,3%
	stability	-	0%	19,6%

Results MDLText

- best in general accuracy
- many datasets (7-44)
- high stability and high accuracy
- more complex

Application area:

- information through context words
  - well-written sentences
- datasets:
  - (medical) science papers (78%<)



evaluation criteria		skip-gram	FastText h=10 2-gram	MDLText,
general	average	57,4%	84,33%	84,8%
	min	35%	60,2%	67,2%
	max	78%	98,6%	98,5%
	stability	43%	38,4%	31,3%
NEWS	average	-	94,65%	84,57%
	min	-	92,5%	67,2%
	max	-	96,8%	92%
	stability	-	4,3%	24,8%
WEB	average	-	98,6%	83,6%
	min	-	98,6%	68,7%
	max	-	98,6%	98,5%
	stability	-	0%	29,8%
word pairs	average	57,4%	-	-
	min	35%	-	-
	max	78%	-	-
	stability	43%	-	-
EN	average	57,5%	79,87%	86,62%
	min	43%	60,2%	67,2%
	max	72%	95,7%	98,5%
	stability	29%	35,5%	31,3%
MULTI	average	-	98,6%	77,69%
	min	-	98,6%	68,7%
	max	-	98,6%	88,3%
	stability	-	0%	19,6%

Conclusion

- skipgram
  - calculation through word pairs
  - good with common subwords in a specific trained area
- fastText + MDLText
  - calculation through text

- low accuracy algorithms can be used in specific fields
- complicated Algorithms are not always better

Thank you for your attention

any questions?

GitHub