Elastic VectorDB and RAG

101 Workshop

Topics for this challenge

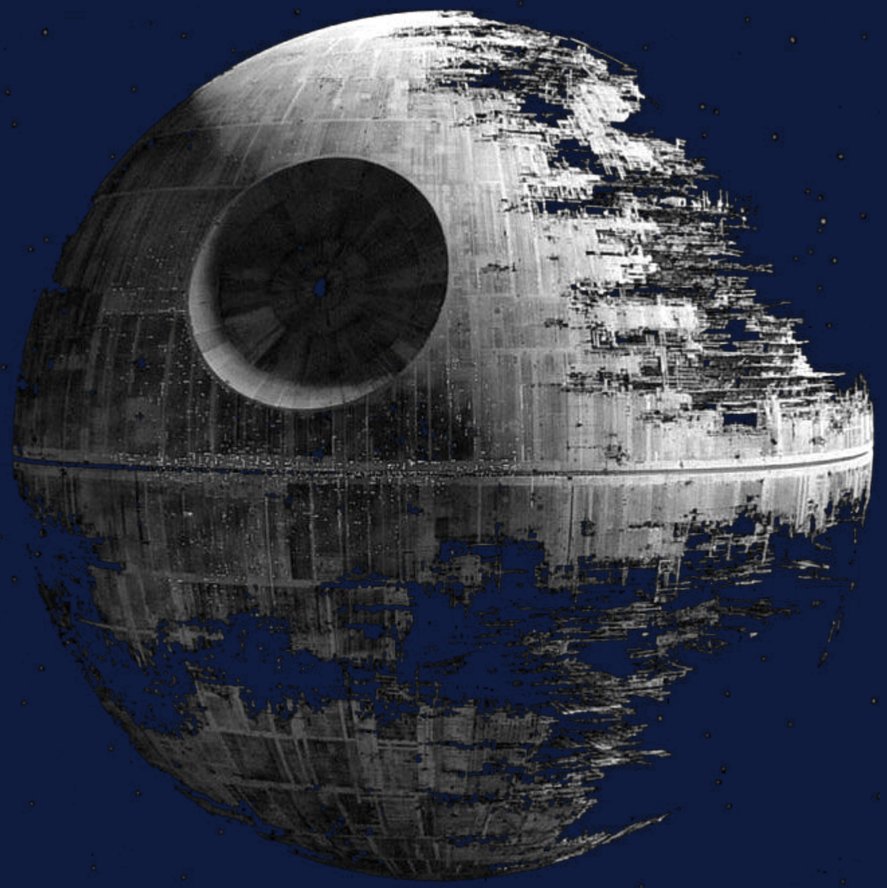
- Vectors
  - Dense and Sparse
  - ELSER
  - Vector Database
  - Chunking
- Inference API
  - Creating the endpoint
- Semantic Text
  - Field type
  - Query type
- RAG
  - Architecture
  - LLMs
    - Brief discussion

elastic

Vectors

Dense and Sparse

Vector embeddings represent meaning



0.0039048328, 0.00070659374, -0.006999771, 0.05141522, -0.005965463, -0.045049522, 0.00019682708, 0.062380187, 0.0110530555,
-0.014826156, 0.026582375, -0.0076479157, 0.07870574, -0.020013824, -0.015210986, -0.03071503, 0.021925598, 0.014036275,
-0.020098573, 0.0013701725, -0.02552182, -0.045320384, 0.023408612, 0.029272491, 0.027291939, 0.027002065, 0.009618439,
0.025841322, -0.03824202, -0.031804346, -0.005024673, 0.019800879, 0.014722629, 0.016817614, 0.0025832115, 0.020656556,
-0.015158481, -0.010246357, -0.021933494, 0.09009692, -0.02062198, -0.0035491597, 0.06875451, -0.014743895, -0.004641175, 0.1214978,
-0.020954289, 4.4564317e-07, 0.002666727, -0.025059653, -0.030262373, -0.008109328, 0.037798755, -0.0077929157, -0.006672016,
-0.051697303, -0.003752414, -0.013640829, -0.024473634, 0.015676884, 0.009082366, 0.01218066, 0.042771876, -0.023839453,
0.032811973, -0.02601853, 0.022658417, -0.03059713, 0.017307144, -0.019683735, -0.026289036, -0.0097188605, -0.03578485,
-0.041527916, -0.010772131, -0.0034551388, 0.003742057, -0.0038169965, 0.029574178, 0.037210364, 0.017164955, -0.0042078975,
0.029943231, -0.012805435, -0.04390587, 0.0044030603, -0.008080069, -0.001404528, -0.010592628, -0.011664122, 0.011565813,
-0.013729068, -0.72748387, -0.0102467965, 0.019818433, 0.04347045, -0.010283088, 0.0041335872, -0.047738794, -0.060791504,
-0.024842288, -0.045538254, 0.038537562, -0.028643234, 0.0009261533, 0.015157209, -0.16175425, 0.011534659, 0.0032795623,
-0.0143686365, -0.034010623, 0.0007185007, -0.043127768, 0.005752832, -0.02827997, -0.022273434, 0.03623347, 0.0072538015,
0.03330577, -0.00047366077, -0.011139977, 0.053952277, -0.02583309, -0.014699725, -0.010210164, 0.014298213, 0.004273705,
0.0050171637, -0.0024857055, 0.028244695, 0.006626626, 0.008594005, 0.004268412, 0.08971255, -0.01349084, -0.033572797,
0.03752266, 0.0024851859, -0.012960998, 0.017297348, -0.012606652, 0.0035163544, -0.021949356, -0.0107436525, 0.022212906,
-0.0051224963, 0.024161108, -0.010141177, -0.0010281279, 0.007809327, 0.036071572, 0.010532015, 0.0030433096, -0.0285774,
0.010959623, 0.0344704, 0.0045044934, 0.025473375, 0.033342265, -0.018964633, 0.033940934, -0.054547116, -0.01330938,
-0.0028191651, 0.0041303826, 0.0027917014, 0.0045481725, 0.037298426, -0.00857898, -0.00055672415, 0.0012227953, 0.025450082,
-0.02577025, -0.0323519, 0.0024521837, 0.002824059, -0.15560368, -0.050387457, -0.0039847963, -0.04072362, -0.00971576, 0.03342038,
0.03261318, 0.011251355, -0.009452698, 0.03529935, -0.01589241, 0.002564077, 0.044983592, -0.0037952578, 0.019388719, 0.011150517,
-0.04457729, -0.0072830818, 0.08218093, -0.005142202, 0.019522818, 0.024081806, 0.010662011, 0.032602258, -0.011701403, 0.01593667,
0.033035316, -0.022232352, 0.02146564, -0.03191656, 0.010158871, -0.035177156, 0.015569976, 0.022177313, -0.00019244145, -0.01269588,
-0.0076705883, -0.03408329, 0.028946845, -0.0077681956, 0.026031248, 0.022365062, -0.005535245, -0.007407689, 0.042693682,
-0.03488027, -0.021181993, 0.01822872, 0.03374636, 0.016557163, 0.0030758402, -0.017818214, 0.053782288, -0.011137253, 0.0022793522,
0.017252814, -0.006168062, 0.03787577, 0.016381413, 0.023566132, 0.03875546, 0.0016214938, -0.012384743, -0.01181817, -0.03858273,
0.00983353, 0.008148084, 0.011728563, 0.033389136, 0.068049595, 0.019063765, -0.022415362, 0.03111486, -0.035040855, 0.015756786,
0.0086081745, 0.026335867, -0.0186453, 0.021732338, 0.042011566, 0.05732202, 0.018153258, -0.096882924, 0.0076769004, -0.038757026,
-0.0012130248, -0.022193724, 0.025630588, -0.0144685805, -0.05468618, -0.005081374, 0.0024184473, 0.019550158, -0.039095078,
-0.034954753, 0.02129486, -0.008054845, 0.015931968, -0.01775497, 0.015053771, -0.011751022, 0.018136406, -0.037537243,
0.0054286625, 0.017992508, 0.02409413, -0.0103469575, -0.008167907, -0.046529993, 0.0081875725, -0.0047017853, -0.0221428,
0.0149085885, 0.029222572, -0.015093489, 0.005506479, -0.038961355, 0.0126315355, 0.015041915, 0.01981507, 0.0054873973,
0.026038013, 0.021696862, 0.04875958, 0.05132017, 0.009879257, -0.027429454, 0.020978231, -0.011259851, -0.00946158, -0.007088396,
0.0068063033, -0.003665152, -0.043090276, -0.028269183, 0.032242734, 0.08976383, -0.006097134, 0.016784793, 0.027979946,
0.05167012, 0.0072538247, -0.018094702, -0.059462022, 0.006947574, 0.030840682, -0.037338357, -0.03376837, 0.0076497807,
-0.006921966, 0.013545439, -0.01011985, 0.020299012, 0.03541653, -0.049154297, 0.032178838, 0.017189631, -0.027203782, 0.034333546,
-0.019677581, 0.03460883, 0.008587687, -0.010567913, 0.0049331724, -0.010977614, 0.032173842, -0.010490562, 0.0062567405,
-0.02630041, -0.010030786, 0.024465112, -0.018689964, -0.0032358554, -0.01137554, -0.014774529, 0.0024110451, 0.013444767,
-0.049109865, -0.05647735, 0.0074918284, -0.04976161, -0.014776791, -0.02945836, -0.004906494, -0.05969401, 0.013291241,
-0.018496571, -0.014744625, -0.101402655, 0.02709476, -0.002597743, 0.0020943764, -0.0003666004, 0.029769192, 0.0009010976,
-0.00096865115, -0.028138407, -0.0272818, 0.01612174, -0.011169845, -0.008978624, 0.03541478, -0.041917346, -0.0049113473,
0.021562282, -0.0018140188, -0.007757695, -0.03720975, 0.037678756, -0.0120101115, 0.04906765, -0.007931659, -0.030482883,
-0.10526867, -0.059618272, -0.010993121, -0.0069628237, -0.03814753, 0.0025131349, -0.00015768249, 0.0015520214, -0.01815863,
0.017955596, -0.07179204, -0.0020325815, -0.011557785, -0.010516109, 0.011058551, 0.005933072, 0.044661947, -0.0090668835,
0.050474122, 0.01110002, -0.05054292, 0.00058114855, 0.014325913, 0.06271102, 0.009368308, -0.026630579, -0.03400157, 0.0013905709,
-0.009244822, -0.026504325, 0.061932158, 0.023238145, -0.024040923, -0.0018142163, -0.00862985, 0.055063576, -0.0088795945,
0.018239843, 0.005045222, 0.06325379, -0.0036330447, 0.06292048, 0.015526667, -0.09055131, -0.0420202, -0.019915584, 0.03653016,
-0.0122567285, 0.025938189, -0.024422025, 0.024138303, -0.027162695, -0.010688026, -0.0044303886, -0.021853466, 0.007994951,
-0.03419652, 0.027959315, 0.044571027, -0.0026074755, -0.0040469756, 0.008565159, 0.019644357, 0.010793187, -0.046515815,
0.035166305, -0.020659845, -0.02326191, -0.035736088, 0.0034593304, -0.026533965, -0.03578475, -0.024412254, 6.6190376e-05,
0.04920903, 0.0071083093, 0.017180137, 0.03999112, 0.012148731, -0.031277765, 0.02001089, -0.036001276, 0.033037607, -0.051119797,
-0.030306648, 0.019238379, -0.019730382, -0.008726098, -0.0028913107, 0.028702024, 0.0016286272, -0.0499603, -0.024412284,
0.04541673, 0.0006619892, -0.021610491, 0.006690292, 0.015699927, -0.017305646, -0.001648571, -0.0041690795, -0.012952357,
0.0013934385, 0.007315853, -0.0138380565, 0.0266797, 0.040392887, 0.0036904349, 0.0349873, -0.019219942, 0.003235542,
0.014629095, 0.022087542, -0.03350893, 0.055818647, -0.02053416, -0.0129638845, -0.02857192, 0.027582409, -0.01811788,
0.00069975335, 0.005343587, -0.021516291

CLIP with ViT-B/32 dense vector representation - 512 dimensions

elastic

Embeddings represent your data

Example: 1-dimensional vector

Character	Vector
	[-1]
	[1]

elastic

Multiple dimensions represent different data aspects

HUMAN

REALISTIC

CARTOON

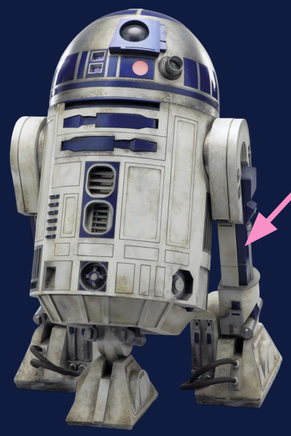
MACHINE

Character	Vector
	[-1, 1]
	[1, 0]

elastic

Similar data is grouped together

Character	Vector
	[-1.0, 1.0]
	[1.0, -0.1]
	[-1.0, 0.8]



HUMAN

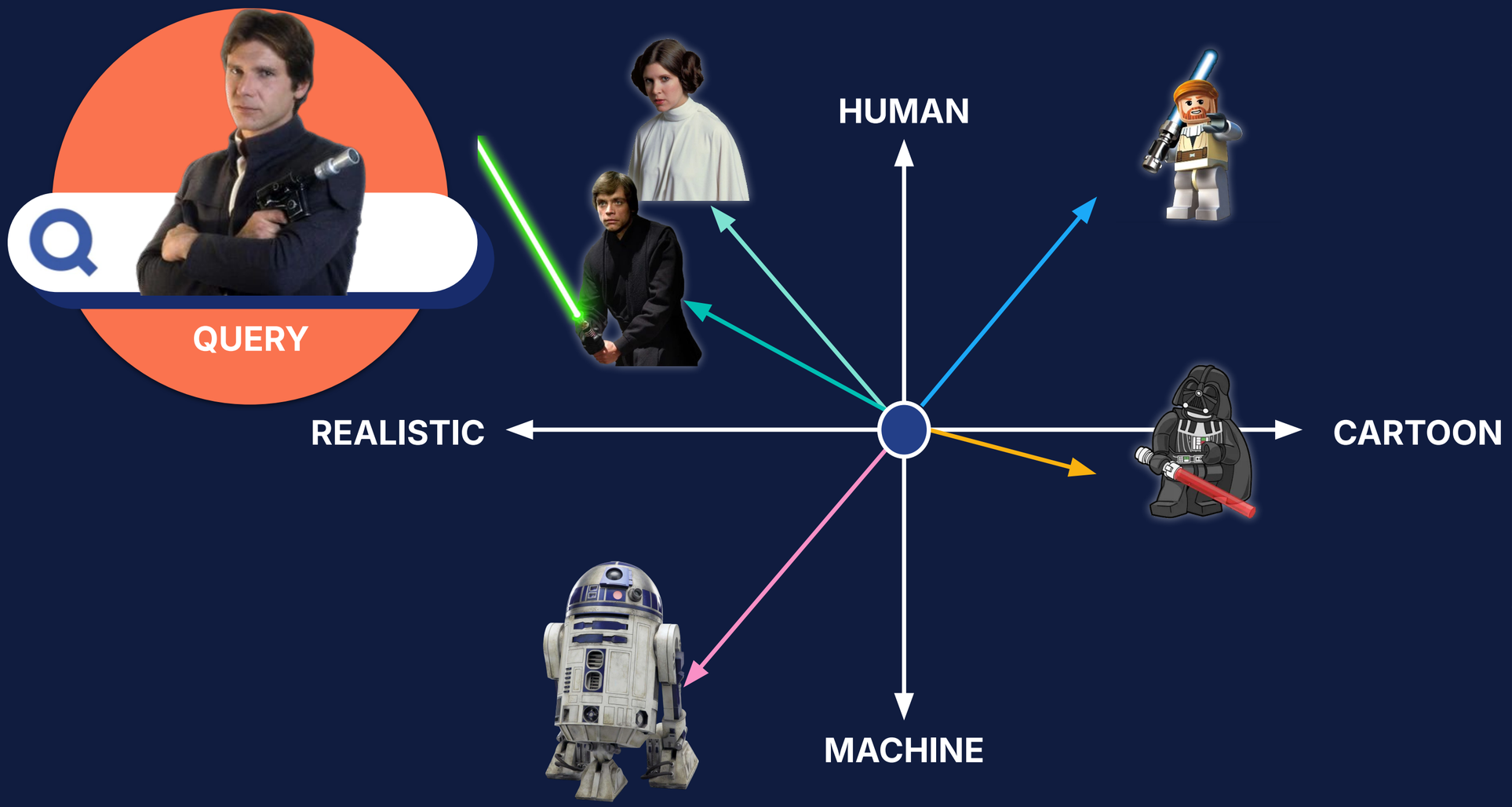
REALISTIC

CARTOON

MACHINE

elastic

Vector search ranks objects by similarity (relevance) to the query



Relevance	Result
Query	
1	
2	
3	
4	
5	

elastic

There are Two Kinds of Vectors Models

Text Input

An image of the Death Star from Star Wars, a large, spherical space station with a gray, metallic surface. It has a distinct circular superlaser dish on its surface, making it instantly recognizable as a powerful weapon from the Galactic Empire

DENSE Vector Output

e5-small

0.0039048328, 0.00070659374, -0.006999771, 0.05141522, -0.005965463, -0.045049522, 0.00019682708, 0.062380187, 0.0110530555, -0.014826156, 0.026582375, -0.0076479157, 0.07870574, -0.020013824, -0.015210986, -0.03071503, 0.021925598, 0.014036275, -0.020098573, 0.0013701725, -0.02552182, -0.045320384, 0.023408612, 0.029272491, 0.027291939, 0.027002065, 0.009618439, 0.025841322, -0.03824202, -0.031804346, -0.005024673, 0.019800879, 0.014722629, 0.016817614, 0.0025832115, 0.020656556, -0.015158481, -0.010246357, -0.021933494, 0.09009692, -0.02062198, -0.0035491597, 0.06875451, -0.014743895, -0.004641175, 0.1214978, -0.020954289, 4.4564317e-07, 0.002666727, -0.025059653, -0.030262373, -0.008109328, 0.037798755, -0.0077929157, -0.006672016, -0.051697303, -0.003752414, -0.013640829, -0.024473634, 0.015676884, 0.009082366, 0.01218066, 0.042771876, -0.023839453, 0.032811973, -0.02601853, 0.022658417, -0.03059713, 0.017307144, -0.019683735, -0.026289036, -0.0097188605, -0.03578485, -0.041527916, -0.010772131, -0.0034551388, 0.003742057, -0.0038169965, 0.029574178, 0.037210364, 0.017164955, -0.0042078975, 0.029943231, -0.012805435, -0.04390587, 0.0044030603, -0.008080069, -0.001404528, -0.010592628, -0.011664122, 0.011565813, -0.013729068, -0.72748387, -0.0102467965, 0.019818433, 0.04347045, -0.010283088, 0.0041335872, -0.047738794, -0.060791504, -0.024842288, -0.045538254, 0.038537562, -0.028643234, 0.0009261533, 0.015157209, -0.16175425, 0.011534659, 0.0032795623, -0.0143686365, -0.034010623, 0.0007185007, -0.043127768, 0.005752832, -0.02827997, -0.022273434, 0.03623347, 0.0072538015, 0.03330577, -0.00047366077, -0.011139977, 0.053952277, -0.02583309, -0.014699725, -0.010210164, 0.014298213, 0.004273705, 0.0050171637, -0.0024857055, 0.028244695, 0.006626626, 0.008594005, 0.004268412, 0.08971255, -0.01349084, -0.033572797, 0.03752266, 0.0024851859, -0.012960998, 0.017297348, -0.012606652, 0.0035163544, -0.021949356, -0.0107436525, 0.022212906, -0.0051224963, 0.024161108, -0.010141177, -0.0010281279, 0.007809327, 0.036071572, 0.010532015, 0.0030433096, -0.0285774, 0.010959623, 0.0344704, 0.0045044934, 0.025473375, 0.033342265, -0.018964633, 0.033940934, -0.054547116, -0.01330938, -0.0028191651, 0.0041303826, 0.0027917014, 0.0045481725, 0.037298426, -0.00857898, -0.00055672415, 0.0012227953, 0.025450082, -0.02577025, -0.0323519, 0.0024521837, 0.002824059, -0.15560368, -0.050387457, -0.0039847963, -0.04072362, -0.00971576, 0.03342038, 0.03261318, 0.011251355, -0.009452698, 0.03529935, -0.01589241, 0.002564077, 0.044983592, -0.0037952578, 0.019388719, 0.011150517, -0.04457729, -0.0072830818, 0.08218093, -0.005142202, 0.019522818, 0.024081806, 0.010662011, 0.032602258, -0.011701403, 0.01593667, 0.033035316, -0.022232352, 0.02146564, -0.03191656, 0.010158871, -0.035177156, 0.015569976, 0.022177313, -0.00019244145, -0.01269588, -0.0076705883, -0.03408329, 0.028946845, -0.0077681956, 0.026031248, 0.022365062, -0.005535245, -0.007407689, 0.042693682, -0.03488027, -0.021181993, 0.01822872, 0.03374636, 0.016557163, 0.0030758402, -0.017818214, 0.053782288, -0.01137253, 0.0022793522, 0.017252814, -0.006168062, 0.03787577, 0.016381413, 0.023566132, 0.03875546, 0.0016214938, -0.012384743, -0.01181817, -0.03858273, 0.00983353, 0.008148084, 0.011728563, 0.033389136, 0.068049595, 0.019063765, -0.022415362, 0.03111486, -0.035040855, 0.015756786, 0.0086081745, 0.026335867, -0.0186453, 0.021732338, 0.042011566, 0.05732202, 0.018153258, -0.096882924, 0.0076769004, -0.038757026, -0.0012130248, -0.022193724, 0.025630588, -0.0144685805, -0.05468618, -0.005081374, 0.0024184473, 0.019550158, -0.039095078, -0.034954753, 0.02129486, -0.0080054845, 0.015931968, -0.01775497, 0.015053771, -0.011751022, 0.018136406, -0.037537243, 0.0054286625, 0.017992508, 0.02409413, -0.0103469575, -0.008167907, -0.046529993, 0.0081875725, -0.0047017853, -0.0221428, 0.0149085885, 0.029222572, -0.015093489, 0.005506479, -0.038961355, 0.0126315355, 0.015041915, 0.01981507, 0.0054873973, 0.026038013, 0.021696862, 0.04875958, 0.05132017, 0.009879257, -0.027429454, 0.020978231, -0.011259851, -0.00946158, -0.007088396, 0.0068063033, -0.003665152, -0.043090276, -0.028269183, 0.032242734, 0.08976383, -0.006097134, 0.016784793, 0.027979946, 0.05167012, 0.0072538247, -0.018094702, -0.059462022, 0.006947574, 0.030840682, -0.037338357, -0.03376837, 0.0076497807, -0.006921966, 0.013545439, -0.01011985, 0.020299012, 0.03541653, -0.049154297, 0.032178838, 0.017189631, -0.027203782, 0.034333546, -0.0196775581, 0.03460883, 0.008587687, -0.010567913, 0.0049331724, -0.010977614, 0.032173842, -0.010490562, 0.0062567405, -0.02630041, -0.010030786, 0.024465112, -0.018689964, -0.0032358554, -0.01137554, -0.014774529, 0.0024110451, 0.013444767, -0.049109865, -0.05647735, 0.0074918284, -0.04976161, -0.014776791, -0.02945836, -0.004906494, -0.05969401, 0.013291241, -0.018496571, -0.014744625, -0.101402655, 0.02709476, -0.002597743, 0.0020943764, -0.0003666004, 0.029769192, 0.0009010976, -0.0009865115, -0.028138407, -0.0272818, 0.01612174, -0.011169845, -0.008978624, 0.03541478, -0.041917346, -0.0049113473, 0.021562282, -0.0018140188, -0.007757695, -0.03720975, 0.037678756, -0.0120101115, 0.04906765, -0.007931659, -0.030482883, -0.10526867, -0.059618272, -0.010993121, -0.0069628237, -0.03814753, 0.0025131349, -0.00015768249, 0.0015520214, -0.01815863, 0.017955596, -0.07179204, -0.0020325815, -0.011557785, -0.010516109, 0.011058551, 0.005933072, 0.044661947, -0.0090668835, 0.050474122, 0.01110002, -0.05054292, 0.00058114855, 0.014325913, 0.06271102, 0.009368308, -0.026630579, -0.03400157, 0.0013905709, -0.009244822, -0.026504325, 0.061932158, 0.023238145, -0.024040923, -0.0018142163, -0.00862985, 0.055063576, -0.0088795945, 0.018239843, 0.005045222, 0.06325379, -0.0036330447, 0.06292048, 0.015526667, -0.09055131, -0.0420202, -0.019915584, 0.03653016, -0.0122567285, 0.025938189, -0.024422025, 0.024138303, -0.027162695, -0.010688026, -0.0044303886, -0.021853466, 0.007994951, -0.03419652, 0.027959315, 0.044571027, -0.0026074755, -0.0040469756, 0.008565159, 0.019644357, 0.010793187, -0.046515815, 0.035166305, -0.020659845, -0.02326191, -0.035736088, 0.0034593304, -0.026533965, -0.03578475, -0.024412254, 6.6190376e-05, 0.04920903, 0.0071083093, 0.017180137, 0.03999112, 0.012148731, -0.031277765, 0.02001089, -0.036001276, 0.033037607, -0.051119797, -0.030306648, 0.019238379, -0.019730382, -0.008726098, -0.0028913107, 0.028702024, 0.0016286272, -0.0499603, -0.024412284, 0.04541673, 0.0006619892, -0.021610491, 0.006690292, 0.015699927, -0.017305646, -0.001648571, -0.0041690795, -0.012952357, 0.0013934385, 0.007315853, -0.0138380565, 0.0266797, 0.040392887, 0.0036904349, 0.0349873, -0.019219942, 0.003235542, 0.014629095, 0.022087542, -0.03350893, 0.055818647, -0.02053416, -0.0129638845, -0.02857192, 0.027582409, -0.01811788, 0.0006975335, 0.005343587, -0.021516291

SPARSE Vector Output

ELSER

death : 2.0881743, star : 1.861367, ##lase : 1.8563019, wars : 1.6361051, circular : 1.4776071, station : 1.4266617, stars : 1.4033791, gray : 1.3953137, metallic : 1.3792193, dish : 1.3677024, surface : 1.3273559, spherical : 1.2689408, weapon : 1.2648199, image : 1.156926, weapons : 1.1151292, large : 1.1123605, space : 1.1061581, powerful : 1.0447476, galaxy : 1.0446026, grey : 0.99213314, recognizable : 0.9624052, picture : 0.83600676, spiral : 0.81533474, alien : 0.80673844, super : 0.78973484, metal : 0.77786785, instantly : 0.7263494, sphere : 0.72603744, laser : 0.72406197, lucas : 0.69926625, mars : 0.6939271, galactic : 0.6567522, rey : 0.64874667, shaped : 0.6404644, surfaces : 0.63722014, shuttle : 0.62602645, empire : 0.61892265, symbol : 0.61714584, radar : 0.6155462, tower : 0.60611147, circle : 0.6031235, iss : 0.6022039, photo : 0.56319225, geometric : 0.55063444, ship : 0.5465214, universe : 0.5411179, images : 0.5366031, gun : 0.53455865, camouflage : 0.53275514, constellation : 0.52575797, familiar : 0.5177047, craft : 0.49535158, type : 0.494465, largest : 0.48790446, shape : 0.4867669, color : 0.4852249, astronomy : 0.4615791, scene : 0.45291653, marvel : 0.4450584, globe : 0.4300228, jedi : 0.41274244, is : 0.41222173, texture : 0.40370402, giant : 0.40316245, terrain : 0.401862, visible : 0.39522073, size : 0.39377856, flight : 0.39123502, plate : 0.37688234, ##r : 0.37562326, exterior : 0.3744266, nasa : 0.37326115, blade : 0.3663758, moon : 0.36008793, triangular : 0.35616356, recognition : 0.35593435, rim : 0.35498098, recognize : 0.35327435, kill : 0.3531336, face : 0.35185418, unique : 0.35141337, ##ase : 0.3484964, telescope : 0.34819987, vast : 0.34729412, spot : 0.346685, appearance : 0.34485716, floating : 0.33843368, ball : 0.3373893, side : 0.3355647, physics : 0.33371758, satellite : 0.33233503, cylindrical : 0.32833597, material : 0.32280168, mobile : 0.32258725, camera : 0.31368616, photograph : 0.3126181, technology : 0.30581737, knife : 0.29537895, point : 0.29529813, stations : 0.29203275, orbit : 0.29163563, rectangular : 0.2890176, illusion : 0.2860448, comet : 0.28443405, lei : 0.2832953, pattern : 0.27580613, bottom : 0.259576, tube : 0.25923607, power : 0.25663304, route : 0.2555368, portrait : 0.25298983, planets : 0.24882926, crescent : 0.24805897, cubic : 0.24250345, instant : 0.23575425, powers : 0.22927032, identification : 0.21873033, commander : 0.21682087, solid : 0.21396767, dishes : 0.21231829, crater : 0.19910999, oval : 0.19613706, missile : 0.19120327, from : 0.19021234, example : 0.18830802, force : 0.18343687, nas : 0.18013021, its : 0.17993645, republic : 0.17644262, war : 0.17062257, ufo : 0.15883592, graphic : 0.15597656, model : 0.15593617, white : 0.15340036, small : 0.15149066, enemies : 0.14789881, atmosphere : 0.13651729, ray : 0.13386005, design : 0.13362469, movies : 0.13128264, fleet : 0.12639168, floor : 0.12328551, gravity : 0.11757267, the : 0.11703412, flat : 0.113113135, category : 0.10692778, cartoon : 0.100848906, earth : 0.093175925, gallery : 0.092740655, robot : 0.092729785, made : 0.08260118, wing : 0.0759074, dye : 0.07394562, probe : 0.065621935, round : 0.060678363, ##sa : 0.057287708, circles : 0.050420593, swords : 0.046301, potent : 0.04328752, origin : 0.043150634, dark : 0.03928177, mission : 0.038609426, cavalier : 0.03609055, core : 0.03511812, 13 : 0.03218187, recognized : 0.030151404, massive : 0.02962671, 3d : 0.028671255, term : 0.025817525, has : 0.02067818, particle : 0.02063299, combat : 0.016088545, cloud : 0.0051144483, tall : 0.004480205, characteristic : 0.0023662446,

elastic

There are Two Kinds of Vectors Models

DENSE Vector

A Long List Of Numbers, one for each dimension

- Trained on a data set for high “In Domain” performance
- Low dimensionality (312, 512, 1536, ..)
- Captures semantic meaning
- Useful for similarity and clustering
- Supports multi-modes
  - Text
  - Image
  - Audio
  - ...
- Memory intensive for larger datasets

SPARSE Vector

Token Weighted Pairs

- Token Vocabulary in the 100’s thousands to millions
- Token Weighted Pairs
  - Token : Weight
- Per document - only N highest weighted tokens stored (the rest are 0’s)
- Semantic search accomplished by DotProduct
- Lower memory requirements compared to dense vector search
- Sparse models can achieve “Late Interaction”

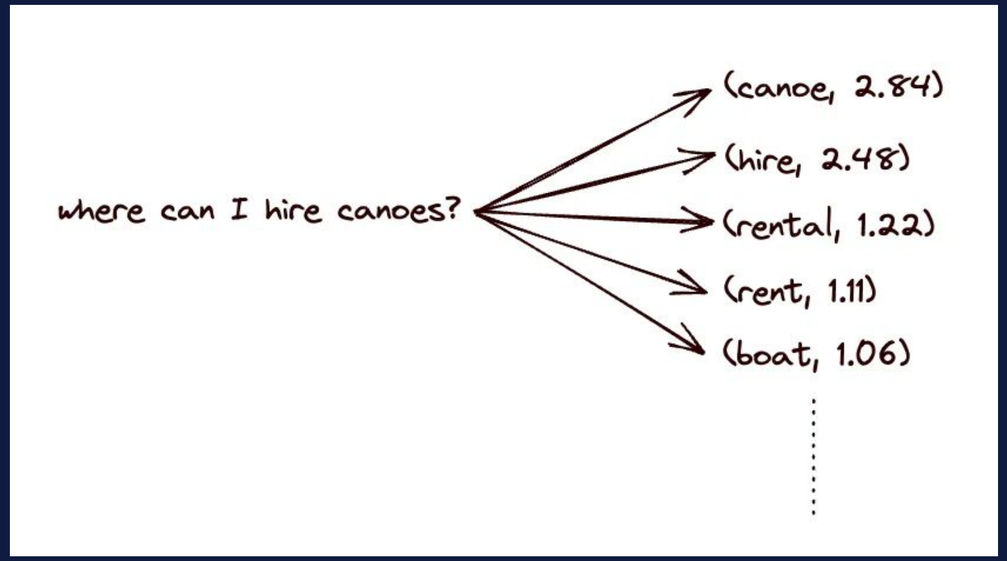
elastic

ELSER

Elastic Learned Sparse EncodeR

ELSER - Elastic Learned Sparse Encoder

- Elastic allows you to bring your own sparse vectors OR use our ELSER models
- ELSER is Commercially licensed transformer model for sparse embeddings
- Inspired by SPLADEv2
- "Text Expansion" model
- “Late Interaction Model”
  - By comparison, dense vectors lose more meaning
  - Scoring happens in expanded token space (late) rather than dense vector
- Blog



elastic

Understanding ELSER

```
POST _ml/trained_models/.elser_model_2_linux-x86_64/_infer
{
  "docs": [{
    "text_field": "These are not the droid's you're looking for"
  }]
}
```


```
{
  "inference_results": [
    {
      "predicted_value": {
        "##oid": 2.0985692,
        "dr": 1.940914,
        "##oids": 1.1135327,
        "looking": 1.0953878,
        "not": 1.0870132,
        "these": 1.0730191,
        "picture": 0.950226,
        "model": 0.9350665,
        "list": 0.8638256,
        "bot": 0.8574795,
        "doctor": 0.8381834,
        "clone": 0.79275626,
        "alien": 0.7516961,
        "galaxy": 0.74790096,
        "robot": 0.7443066,
        "s": 0.72506857,
        "badge": 0.71229017,
        "image": 0.6914874,
        "this": 0.6688425,
        "platform": 0.6652705,
        "for": 0.6276486,
        "cartoon": 0.62331337,
        "are": 0.61874485,
        "models": 0.6000352,
        "monster": 0.5642273,
        "mascot": 0.55157906,
```


elastic

Text Expansion - Not synonyms

Stored in Elastic

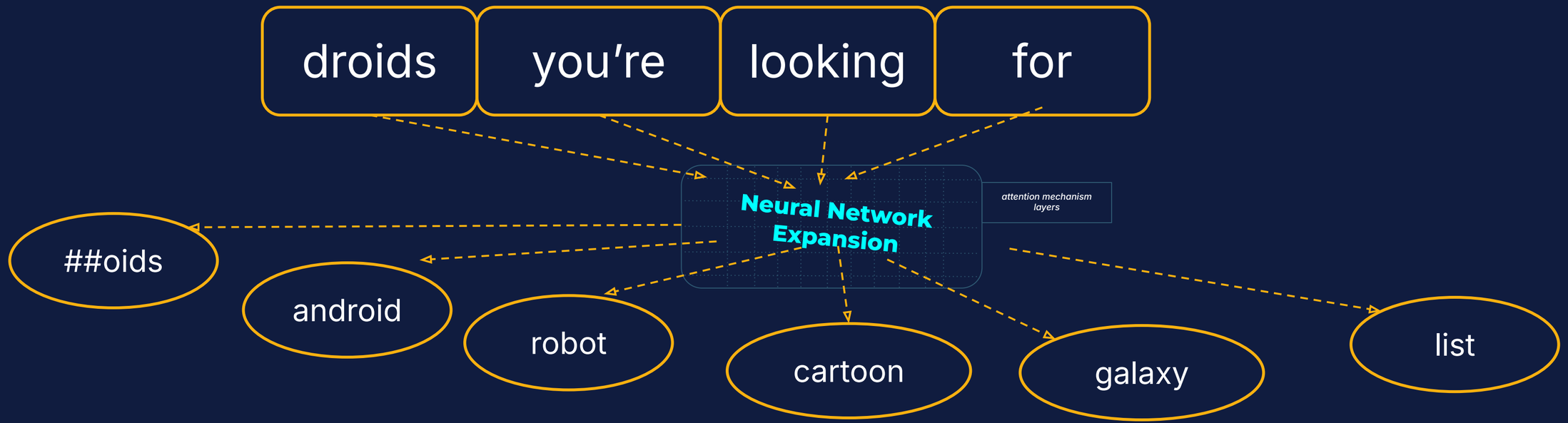
droids	you're	looking	for

Input Query

elastic

Text Expansion - Not synonyms

Stored in Elastic

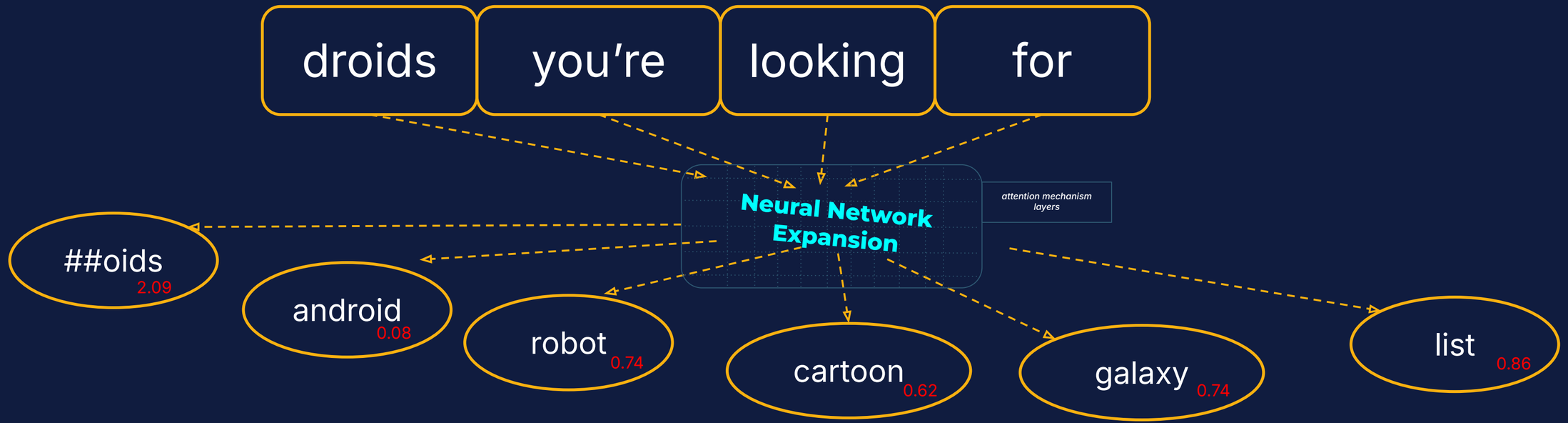


Input Query

elastic

Text Expansion + Weights

Stored in Elastic



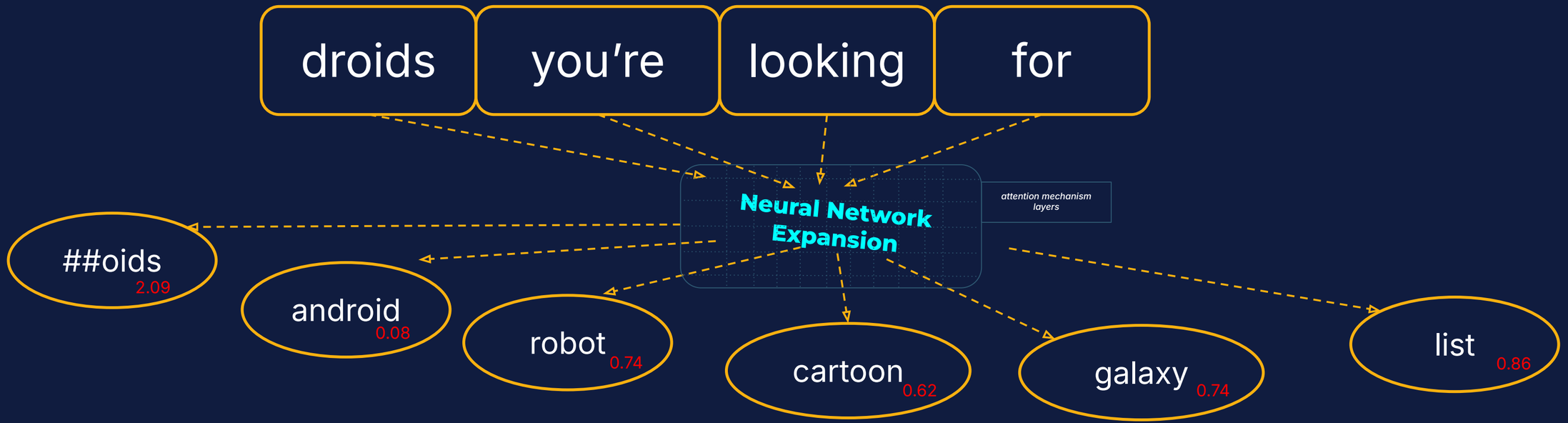
Input Query

elastic

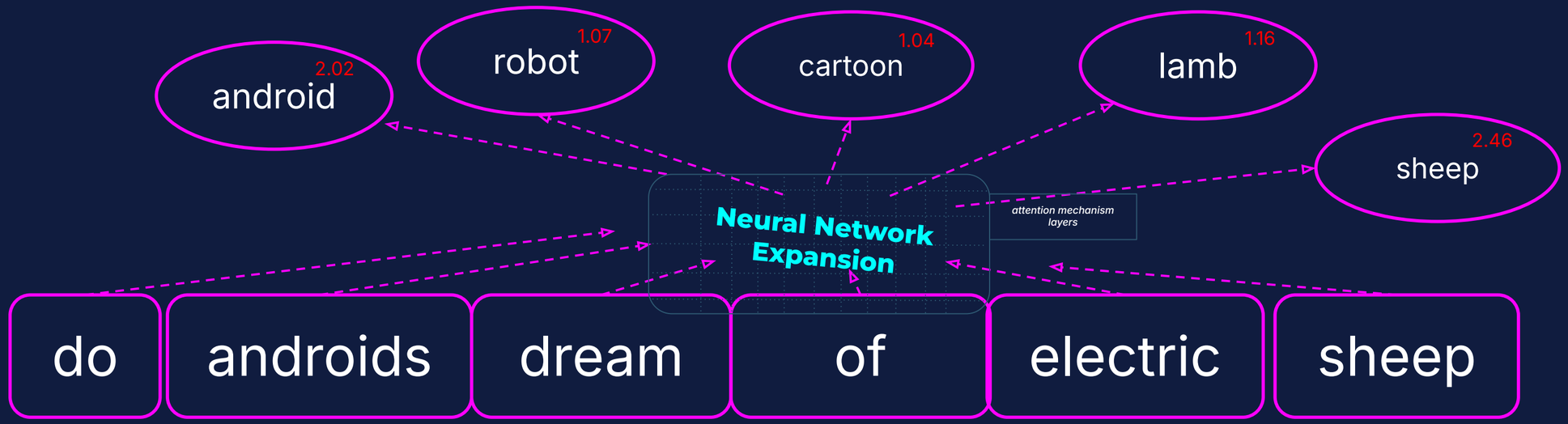
Text Expansion + Scoring

with Sparse vector query

Stored in Elastic



Input Query

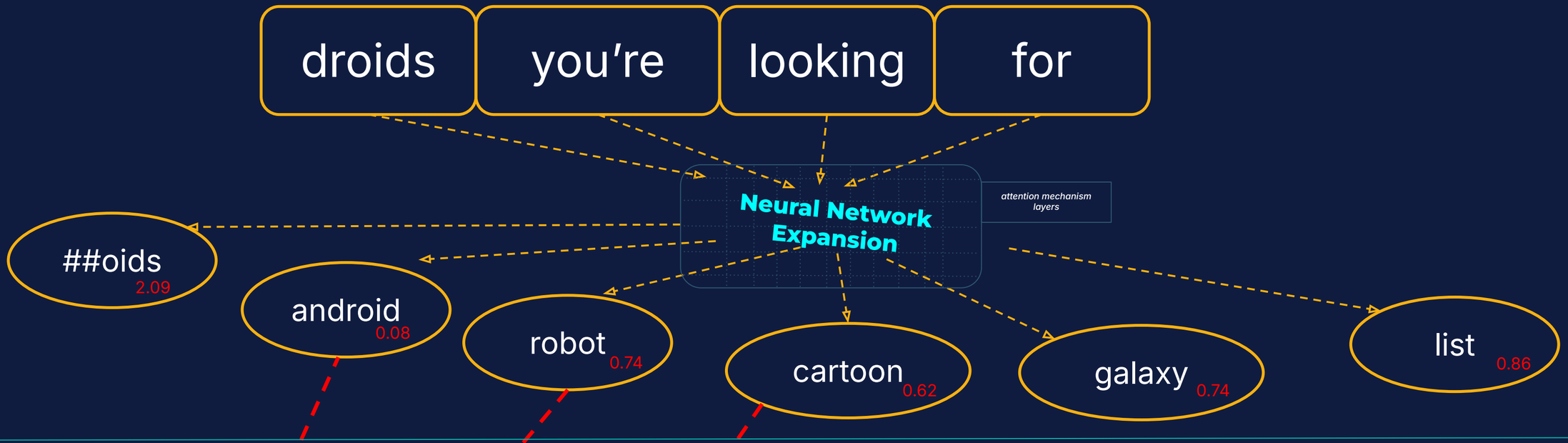


elastic

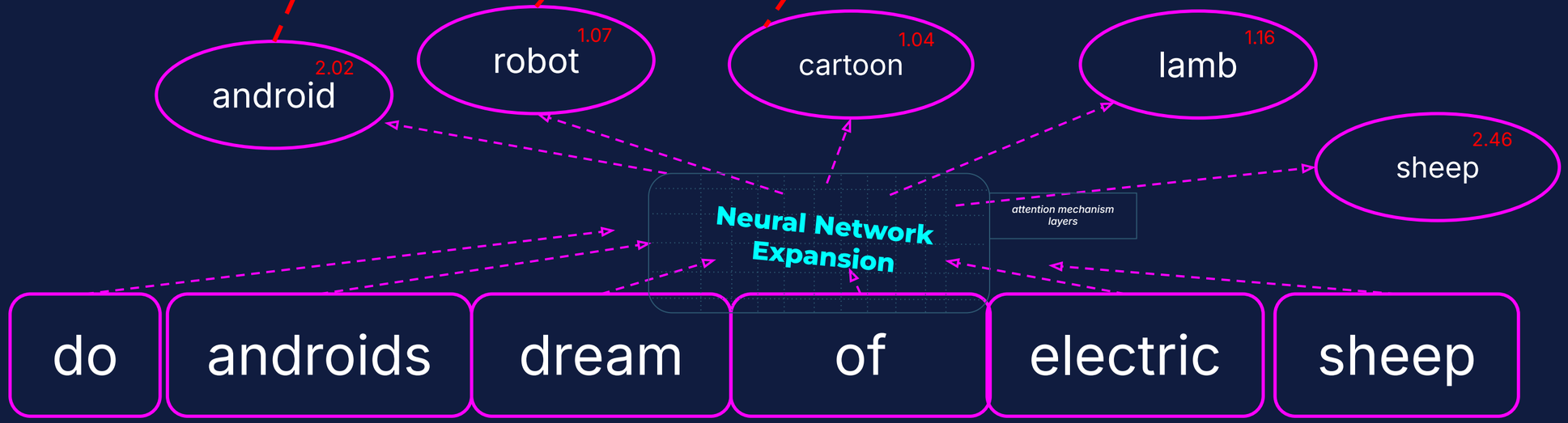
Text Expansion + Scoring

with Sparse vector query

Stored in Elastic



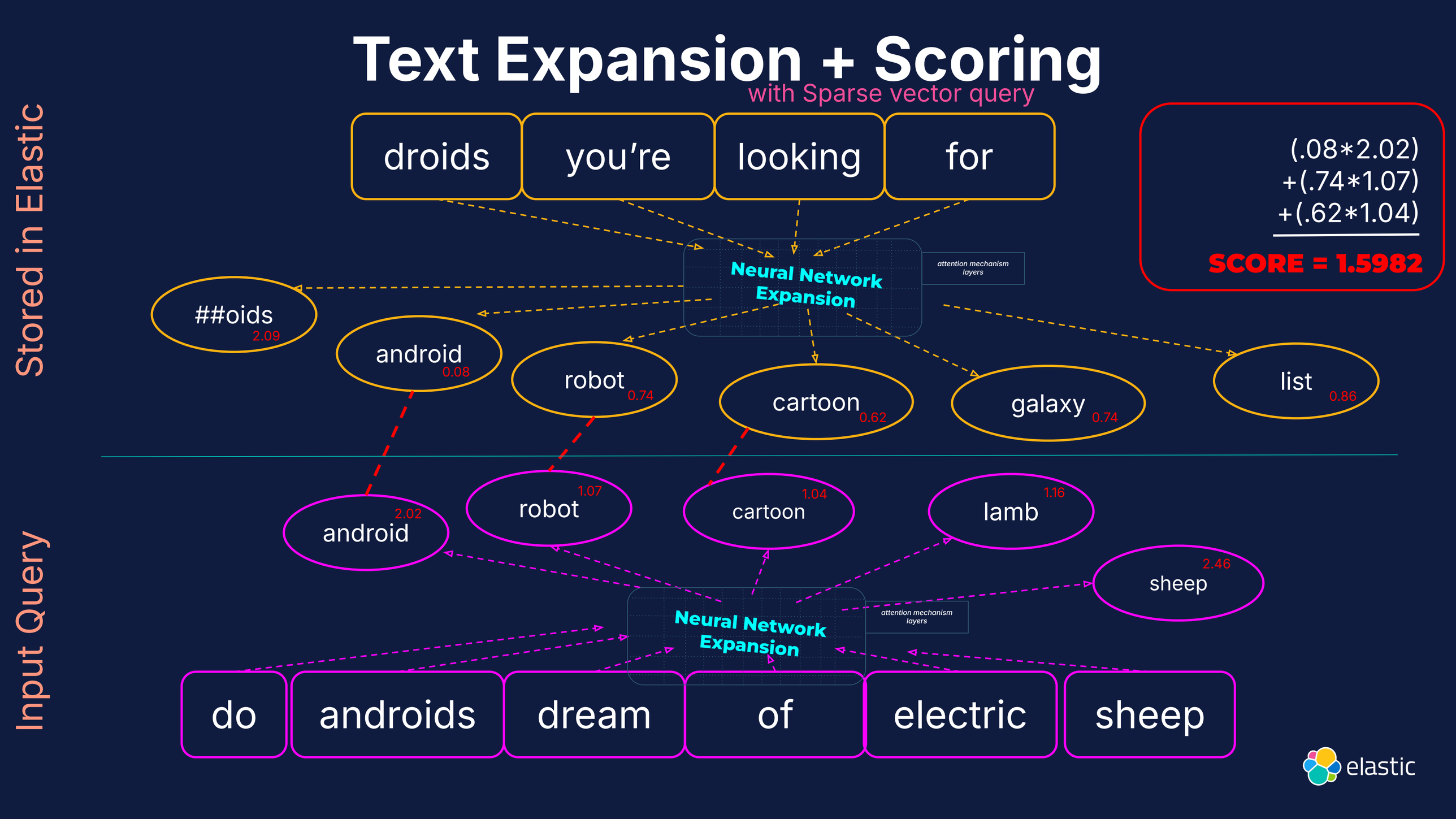
Input Query



elastic

Text Expansion + Scoring

with Sparse vector query



elastic

ELSER - Elastic Learned Sparse Encoder

- Currently on ELSERv2
  - Elastic comes an ELSERv2 optimized for x86 instruction set
  - Inference currently occurs on CPU only
- Efficient
  - Utilizes Lucene native indices at query time
  - Extremely storage efficient vs. ColBERT (w/ vector per token)
- Quality
  - High performance on BEIR in “out of domain test”
  - Comparable to higher performance than SPLADEv2 and ColBERT
- Blog

Full BEIR results

BEIR subset score (OpenAI does not publish vs. vull BEIR)

	Vespa (colBERT)	Instructor	SPLADEv2	ELSER	OpenAI (davinci)	OpenAI (ada)	ELSER
Average	0.452	0.441	0.47	0.486	0.528	0.533	0.538

Scores from ELSERv1 from 2023

elastic

ELSER vs Open Transformer Models

Search results ranking models ↓ — Data sets (BEIR benchmark) →

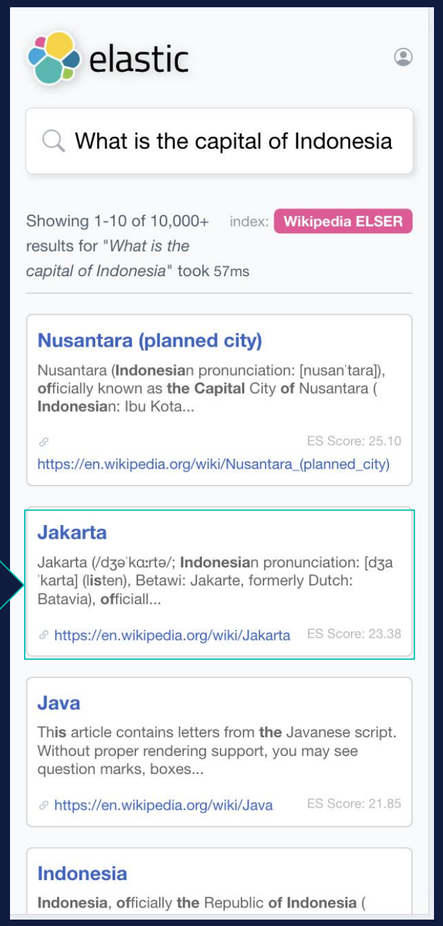
	Average	TREC-COVID	NFCorpus	NQ	HotpotQA	FiQA	ArguAna	Touche-2020	DBPedia	SCIDOCS	FEVER	Climate-FEVER	SciFact
BM25	0.416	0.688	0.327	0.326	0.602	0.254	0.472	0.347	0.287	0.165	0.649	0.186	0.69
RRF (BM25/Dense)	0.449	0.697	0.317	0.445	0.611	0.318	0.474	**0.354**	0.353	0.159	0.746	0.238	0.671
Linear (BM25/Dense)	0.471	0.787	0.335	0.485	0.62	0.341	0.444	0.346	0.378	0.164	**0.778**	**0.272**	0.698
SPLADE	N/A	0.726	0.347	**0.537**	0.687	**0.347**	**0.526**	0.246	**0.436**	0.158			0.703
Elastic Learned Sparse Encoder	0.471	0.747	**0.351**	0.524	0.67	0.339	0.5	0.263	0.415	0.156	0.777	0.218	0.695
RRF (BM25 / Elastic Learned Sparse Encoder)	**0.478**	**0.797**	**0.352**	0.468	0.674	0.311	0.497	0.347	0.411	**0.166**	0.762	0.24	**0.712**

Search quality *worse* than benchmark (BM25) — Search quality *better* than benchmark (BM25)

- Out-performs on: question-answer pairs, weather records, medical text
- Generalises across domains without training
- Exceptional results without need for Fine Tuning per data set

ELSER

Elastic sparse model



E5

Open-source Model

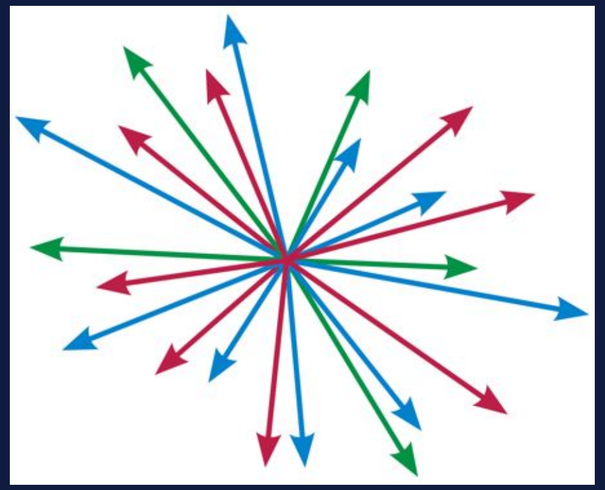
elastic

VectorDB

What is a Vector Database?

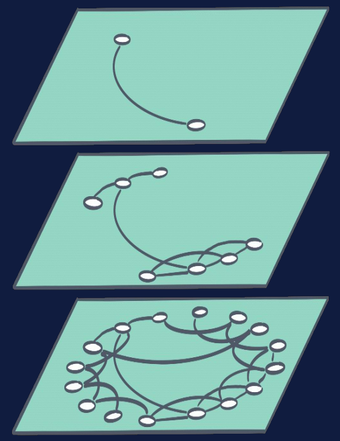
ARE YOU INDEXING FOR MEANING?

1) Stores "**Meaning**" of Text, Images, and Audio encoded in high dimensional dense and sparse vectors



2) Indexes for quick and efficient **vector search** at massive scale.

Approximate Nearest Neighbor (ANN)



Elastic utilizes an optimized HNSW data structure

3) Provides must-have features for **production** use cases

Deploy Anywhere

Security & Multi-Tenancy

Index-Intersection (i.e. **temporal**, geospatial)

Cross-Region Search

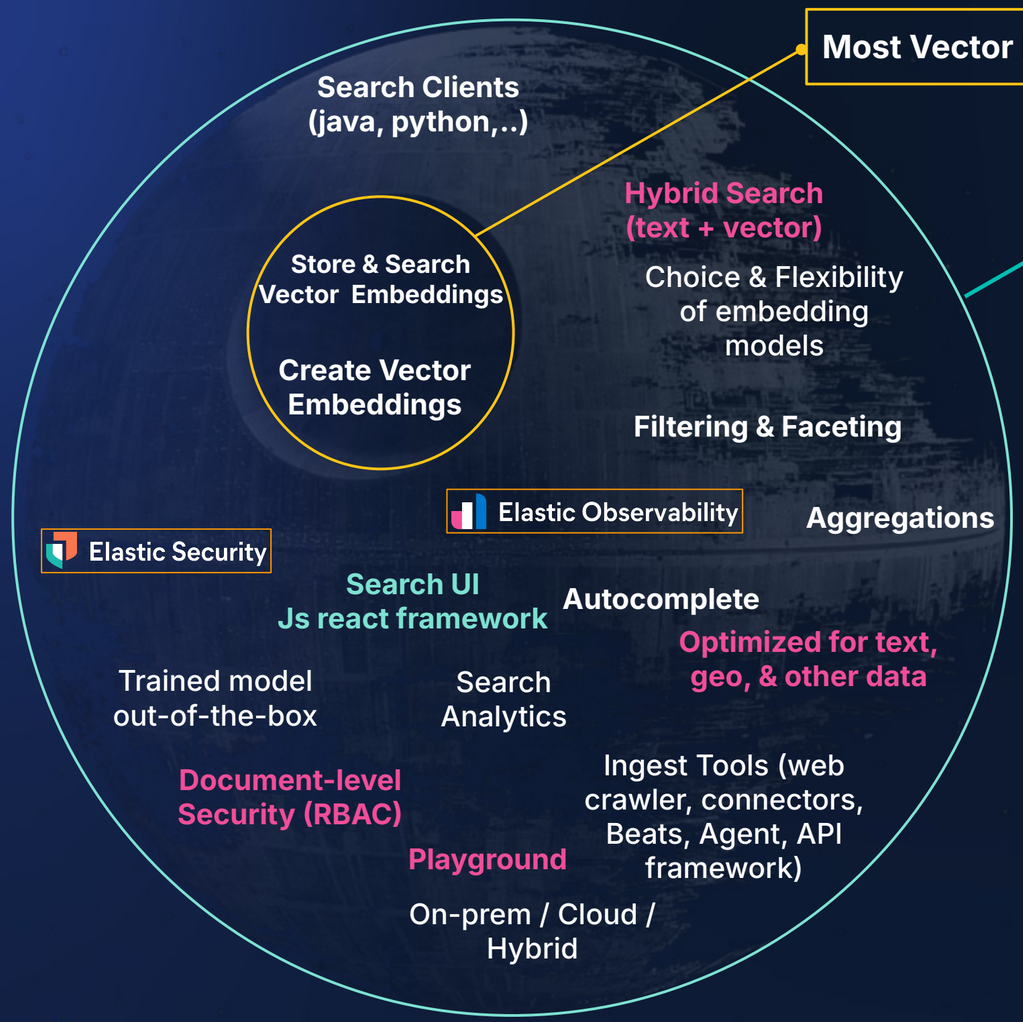
Reranking

HA/DR

... many more

elastic

Elasticsearch - A VectorDB and Much More



Elasticsearch provides the full scope of necessary capabilities for Search & Generative AI applications, beyond anyone else

Elasticsearch

- **Vector database**
  Fast, Memory optimized, int8, int4 quantization
- **ELSER**
  Proprietary Zero shot out of domain semantic search model
- **Full-Text search engine**
  Powered by latest Lucene
- **RRF for hybrid scoring - Reranking**
  Vector & textual search, Increased Relevance
- **Ability to host your own transformer model**
  Save money & time
- **Ability to integrate with 3rd party transformer models (OpenAI, Cohere, Hugging Face)**
  Flexible, Do more with Elastic
- **Integration with 3rd party tooling like LangChain**
  Do with your preferred framework, the Elasticsearch Class of which we maintain

Computing Vectors should be **simple** and **flexible**

Elasticsearch can:

1. use vectors computed outside Elasticsearch (they are just JSON)
2. host your pytorch vector model for you
3. call your favorite cloud hosted vector inference apis (OpenAI, Cohere, HuggingFace, etc)

Elastic comes out of the box with two great models

DENSE VECTOR

E5

Multi-Lingual

SPARSE VECTOR

ELSER

(Elastic Learned Sparse Encoder)

elastic

Chunking

Long Input Texts

What is Chunking?
Why Chunk?
How to Chunk?

What:

- The process of splitting text into smaller pieces (chunks)
  - Used with longer passages of text

Why:

- Embedding models have context limit for input
  - Usually 512 tokens
  - 1 token ~= ¾ of a word
- Tokens that exceed the input limit will be dropped

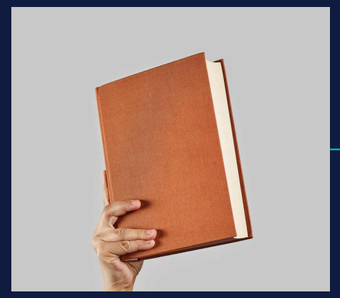
How:

- Before sending to Elasticsearch
  - Custom script
  - Library tools (eg. Langchain)
- With a painless scripts on ingest to Elasticsearch
  - Fully customizable but requires some scripting
  - Requires ingest pipeline configuration
- Using `semantic_text` field in Elasticsearch
  - Automatic chunking on ingest

elastic

A few examples of

Chunking strategies



Paragraph

Sentence

Word

Vector models are limited in how many tokens they can take as input.

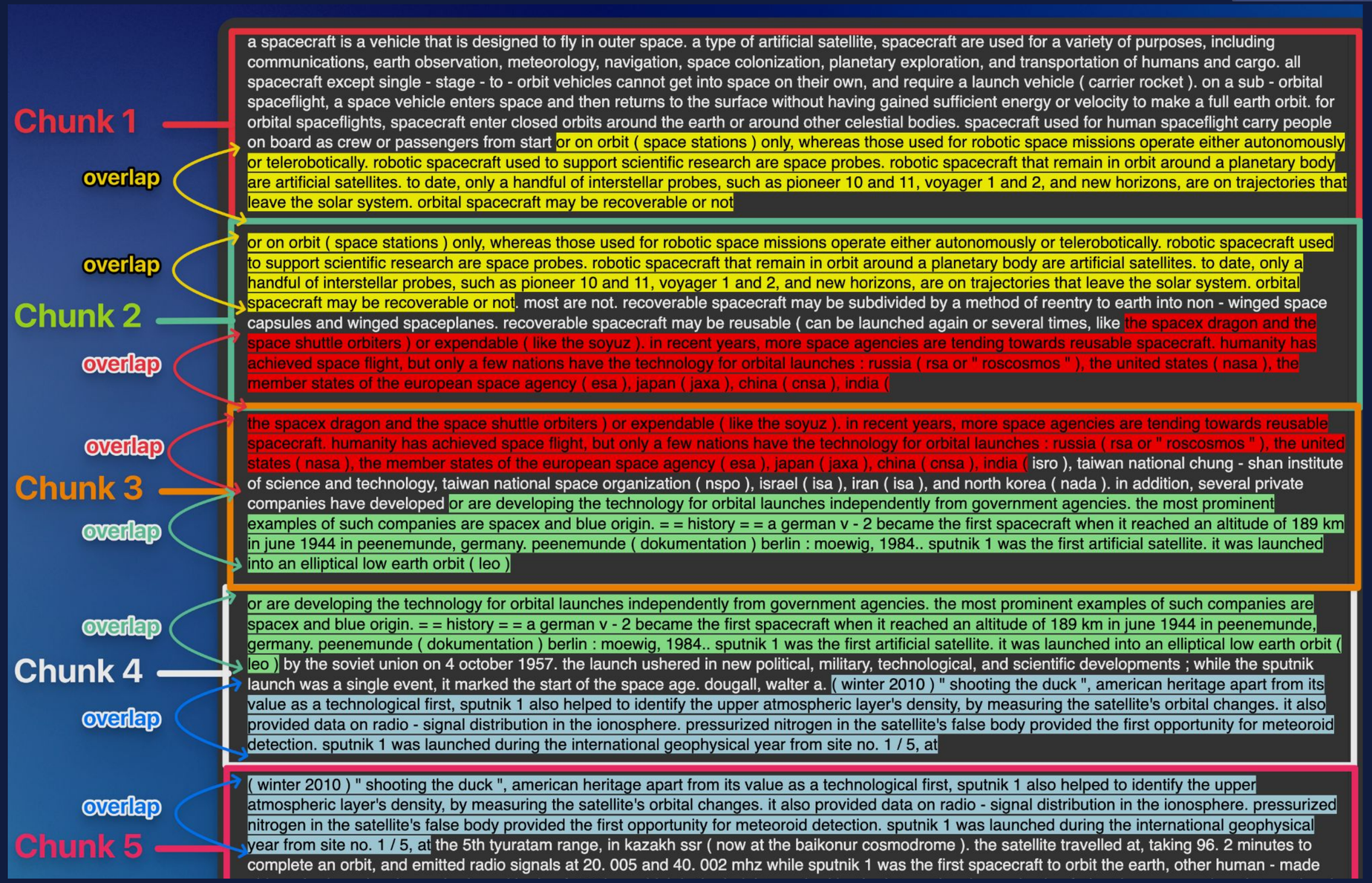
Most truncate after the first X tokens ... dropping meaning

The smaller the chunk, the less likely a single chunk can hold all the information necessary to answer a question

elastic

Chunking in Elasticsearch with `semantic_text` Automatically

- Text is split into 250 word chunks
- Each chunk includes 100 words from the previous chunks
- Chunk passed to Inference API for embedding
- Original text stored along with chunks, chunk vectors
- Using `semantic` query will automatically search nested chunks in a `semantic_text` field.



Inference API

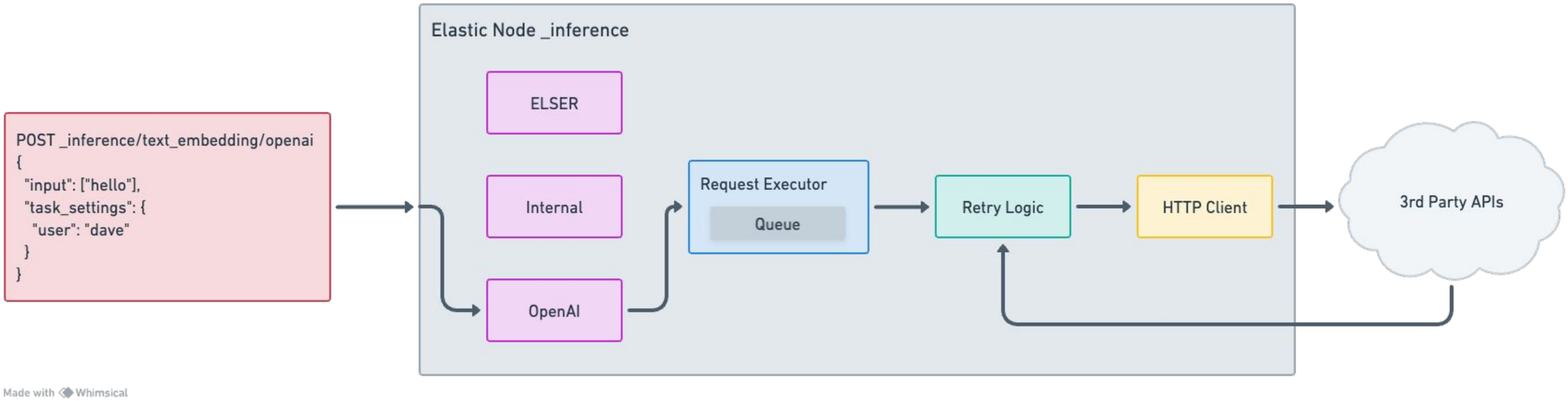
Creating Endpoints

Inference API Endpoint

- Support for the built in models
  - E5 & ELSER
- External Text Embedding Services
  - Cohere
  - OpenAI
  - Azure OpenAI & AI Studio
  - AWS Bedrock
  - Google Vertex AI & AI Studio
  - Hugging Face

elastic

API Flow



elastic

Semantic Text

The Easy On-Ramp

Semantic Text uses the Inference API

- Out of the box Config
- Auto-Chunking long documents
- Ingestion & query embeddings
- Automatic Chunking
- Inference / model co-ordination
- Automatically generates embeddings for text
- Don’t need to specify how to generate embeddings
- Automatically determines the embedding generation, indexing, and query to use

Field Mapping And Querying

Semantic Text Field Mapping

```
"my_semantic_field": {
    "type": "semantic_text",
    "inference_id": "test-elser",
}
```

Semantic Query

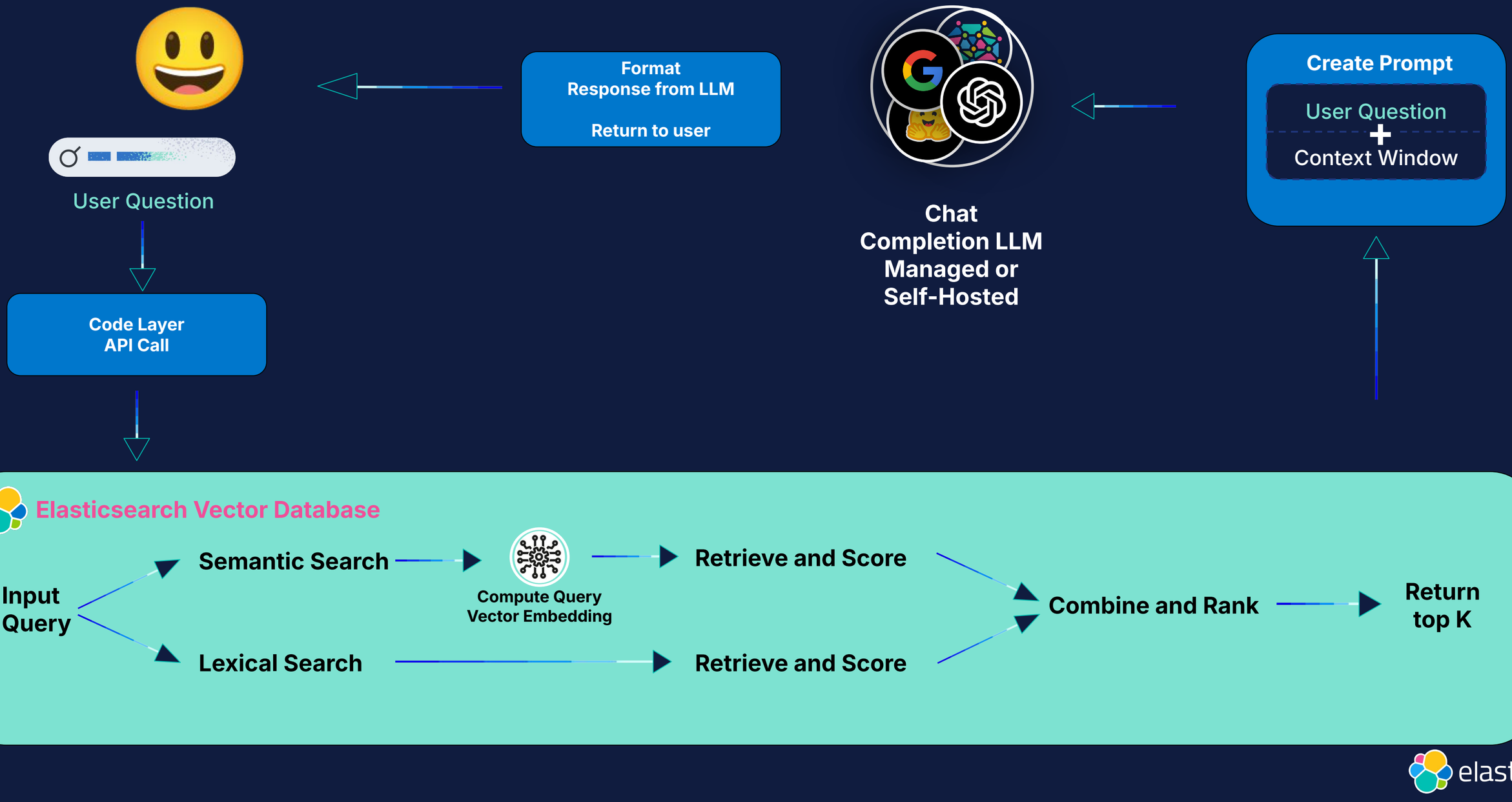
```
POST test-index/_search
{ "query": {
    "semantic": {
      "field": "my_semantic_field",
      "query": "robots you're searching for"
    } } }
```

elastic

RAG

Retrieval Augmented Generation

Retrieval Augmented Generation : How it works



elastic

Example: Understanding intent in Customer Experience

"OMG I can't find my insurance card, I'm at the doctor's office right now, what should I do?"

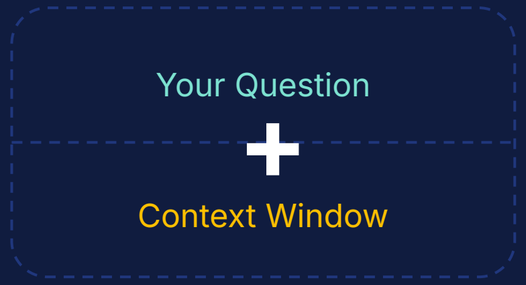

```
{
   vector : [0.1234, 0.5678, …. ],

   question : "At the point of care when a
   patient does not have their proof of
   membership …",

   solution : "suggest the user utilize their
   mobile app for proof of coverage …",

   metadata : {…}
}
```

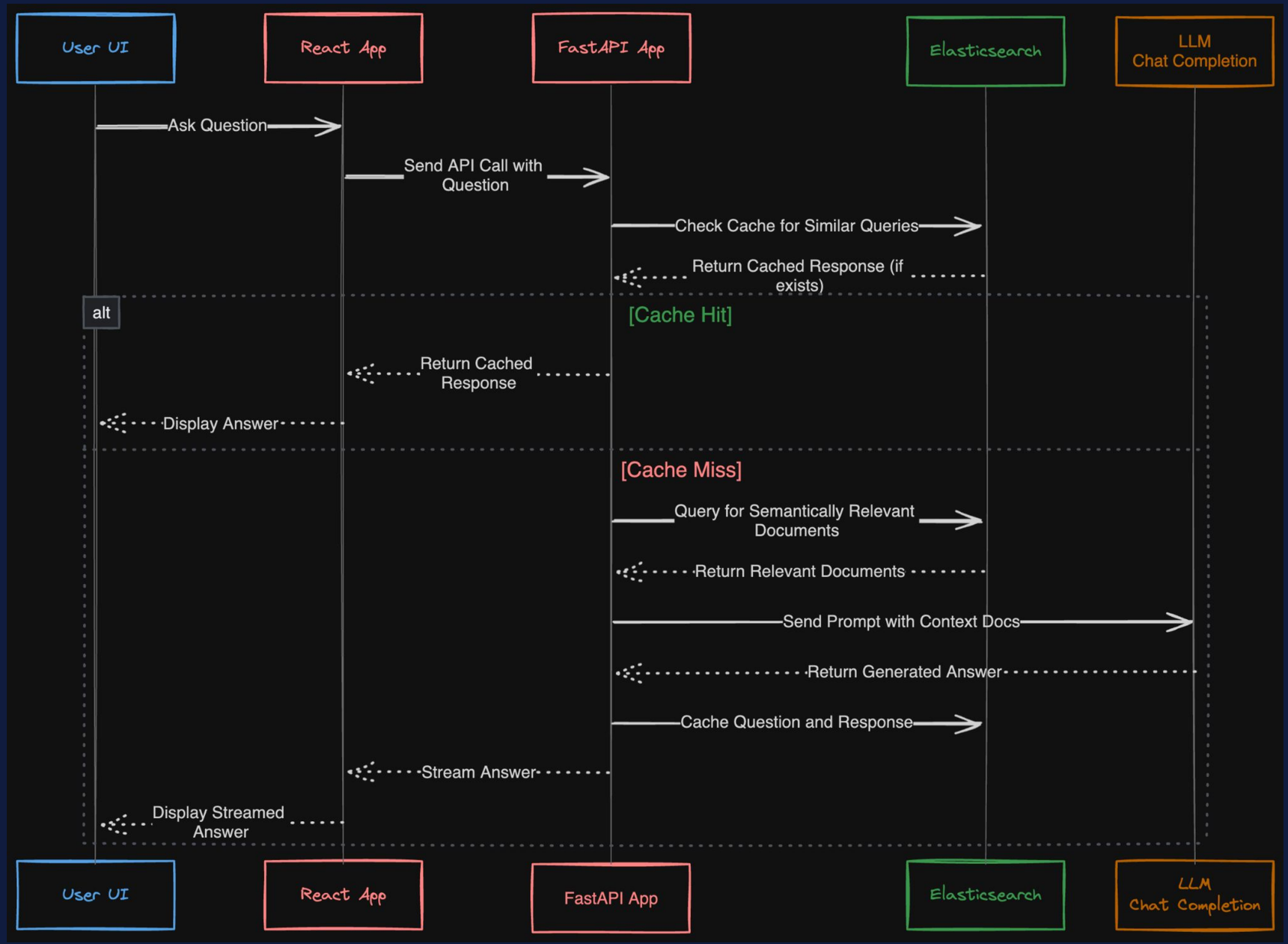
Assemble the prompt for LLM



LLM: "I can see that you use our mobile app. Open that up and get your card by navigating to My account > Proof of Insurance."

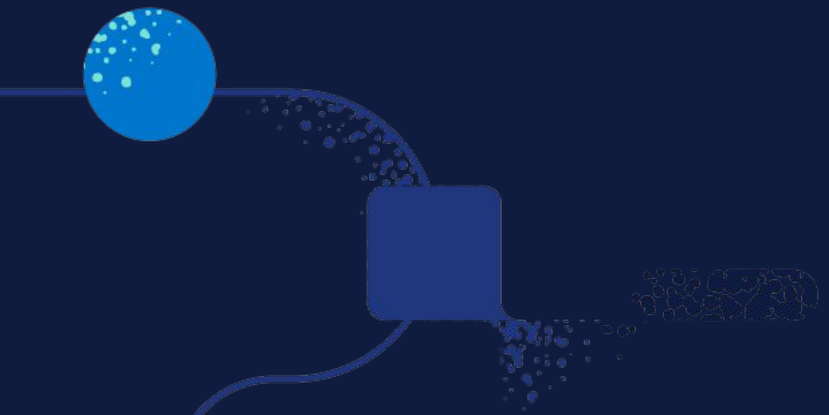
elastic

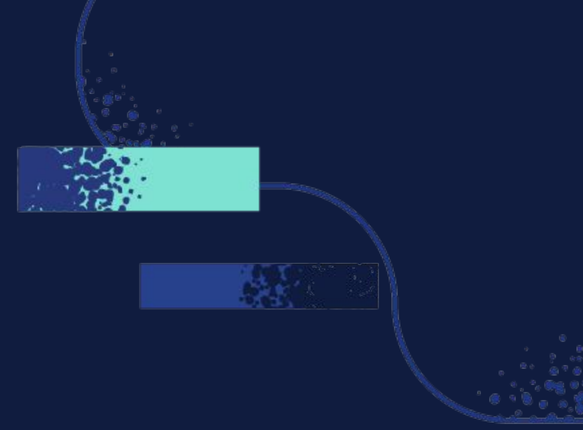
RAG Architecture and Data Flow



elastic

End





elastic