

Greedy decoding: 현재까지 가장 좋은 것만 선택

(-) 효율적이지 않다.

Exhaustive search: 임의의 문장은 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

$$P(y_1, y_2, \dots, y_n) = P(y_1 | z) P(y_2 | y_1, z) P(y_3 | y_1, y_2, z) \dots P(y_n | y_1, \dots, y_{n-1}, z)$$

주어진 문장에 대한 확률값을 계산하기 위해

→ 모든 가능한 길은 계산

(-) V가 vocabulary size이면 V의 제곱 수가 되는데

$O(V^2)$ 의 시간 복잡도는 너무 expensive이다

Beam search

core idea: 이 time step에서, <start> 또는 이후 등장한 단어들의 점수는 모두 고려하여 가장 높은 k개의 후보만 선택한다

여기서 k는 beam size (일반적으로 5~10)

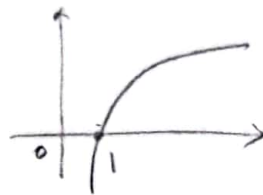
k개의 제곱 수가 되어 decoding output은 k개의 hypothesis라 한다

$$\text{Score}(y_1, y_2, \dots, y_n) = \log P_{\text{LM}}(y_1, y_2, \dots, y_n | z) = \sum_{i=1}^n \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, z)$$

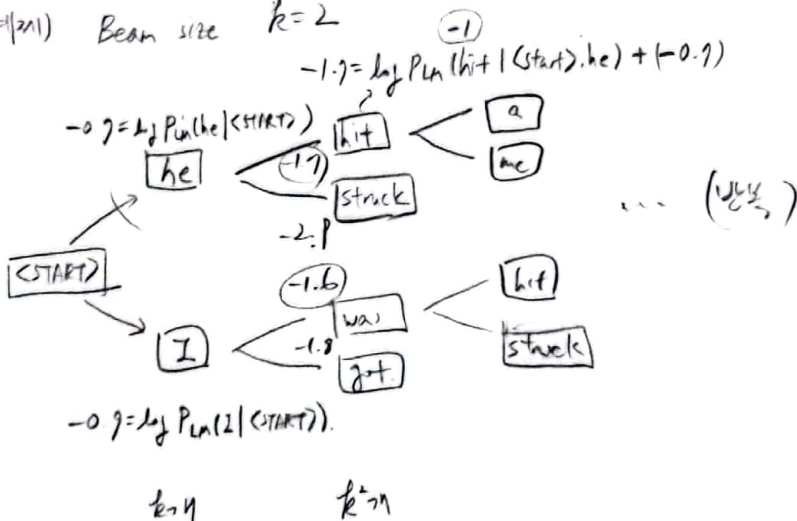
클수록 좋다. (log: 단조증가), Score는 모두 음수

(-) 가장 좋은 것만 찾는 게 아니다

(+) Exhaustive search 보다 효율적



예제1) Beam size $k=2$



greedy decoding과는 다르게 서로 다른 hypothesis에서 <end> token이 나올 수 있음

<end> token이 나오면 hypothesis가 완료된 것으로 본다, 임의의 hypothesis에 해당

여기서 선택한 cut-off 가장 높은 Token을 중단하지만, 선택된 hypothesis n개에서 중단

Beam search가 끝나는 순간의 hypothesis의 top k를 선택한다
 여기서 가장 높은 score를 갖는 top k개의 hypothesis를 뽑는다.

$$\text{Score}(y_1, \dots, y_t) = \log \text{P}_{\text{LM}}(y_1, \dots, y_t | x) = \sum_{i=1}^t \log \text{P}_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$

문제를 킬 hypothesis는 상대적으로 작아져 버린다

이를 보완하기 위해, 각 hypothesis 별로 단어 개수를 나눠서 평균으로 구한다

$$\text{Score}(y_1, \dots, y_t) = \left(\frac{1}{t} \sum_{i=1}^t \log \text{P}_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x) \right)$$

평가 지표

Reference : Half of my heart is in Havana ooh na na
 (Ground-Truth)

predicted : (Half) as my heart (is) (in) Obama (ooh) (na)

$$\text{precision} = \frac{\text{정답 단어 수}}{\text{Predicted 단어 수}} = \frac{7}{9} = 78\%$$

$$\text{Recall} = \frac{\text{정답 단어 수}}{\text{Reference 단어 수}} = \frac{7}{10} = 70\%$$

} 일치한 단어

$$\text{F-measure} = \frac{\text{precision} \times \text{recall}}{\frac{1}{2} (\text{precision} + \text{recall})} = \frac{0.78 \times 0.7}{0.5 \times (0.78 + 0.7)} = 73.78\%$$

f) 정확도 \geq 기하평균 \geq 재현율.

예제) Predicted (Model 1) : (Half) as my heart (is) (in) Obama ooh na

Reference : Half of my heart is in Havana ooh na na

Predicted (Model 2) : Havana na in heart my is Half ooh of na

	Model 1	Model 2
precision	78%	100%
recall	70%	100%
F-measure	73.78%	100%

↓
정답 단어 수

(-) 재현율에 비해 정확도가 낮음

BLEU (Bilingual Evaluation Understudy)

N-gram overlap의 정도를 평가 (predicted와 reference 간)

1-gram과 4-grams 사이의 precision 계산 (recall은 무시) → 2개의 평가가 더 크고

brevity penalty를 사용, 번역이 너무 짧은 것의 패널티를 준다

$$BLEU = \min\left(1, \frac{\text{length of precision}}{\text{length of reference}}\right) \left(\prod_{i=1}^4 \text{precision}_i\right)^{\frac{1}{4}} \leftarrow \text{기하평균}$$

일반적으로 전체 corpus에 대해서 수행한다 (단, 문장 자체만 수행하는 것이 있다)
→ 번역이 정확하면, 평가할 수 있는 recall이 높아진다.

ex) Predicted (Model 1) : Half as my heart is in Obama ooh na
Reference : Half of my heart is in Havana ooh na na
Predicted (Model 2) : Havana na in heart my is Half ooh of na

	Model 1	Model 2
precision (1-gram)	7/9	1/10
precision (2-gram)	4/8	0/9
precision (3-gram)	2/7	0/8
precision (4-gram)	1/6	0/7
Brevity penalty	8/10	10/10
BLEU	$0.9 \times \sqrt[4]{\frac{1}{54}} \approx 33\%$	0