NLP HW3

Tom Barzilay, Gilad Vardy-zer, Yonatan Tintpulver

June 2023

1 Question 1

1.1

***a*)** We will point out 3 important characteristics needed:
* α_i is non negative for each i. This is because the softmax function uses exponents, meaning that the numerator and denominator are both always non negative.
* $\sum \alpha_i = \sum_{i=1}^{n} \frac{exp(k_i^T q)}{\sum_{j=1}^{n} exp(k_j^T q)} = \frac{1}{\sum_{j=1}^{n} exp(k_j^T q)} * \sum_{i=1}^{n} exp(k_i^T q) = \frac{\sum_{i=1}^{n} exp(k_i^T q)}{\sum_{j=1}^{n} exp(k_j^T q)} = 1$
the denominator can be taken out of the sum since it is a multiplication. We got that the sum of all the probabilities is 1.
* The probability of each category is separately specified. Meaning that for each i, we get a probability that matches that specific category.
Since all 3 properties exist, we get the α can be interpreted as a categorical probability distribution.
***b*)** This can happen if the dot product between k_i^T and q is larger than all other keys, this can happen (for instance) if the query is the same (or really similar) as a specific key.
***c*)** Under the conditions in question b, we will get a value c that is closely related to that of α_i, this is because α_i is the most dominant and so the overall sum of α is mostly influenced by that specific α_i. The other α_j (matching the other keys) will change the sum, but at a much smaller extent. In all we get a c that almost matches α_i.
***d*)** Intuitively, this means that we got c that is almost identical to a value vector (we "copied" it).

1.2

***a*)** let T be T = $\begin{bmatrix} | & | & | & | & | & | & | & | \\ a_1 & a_2 & ... & a_m & b_1 & b_2 & ... & b_p \\ | & | & | & | & | & | & | & | \end{bmatrix}$ such that the columns of this matrix are the vectors $\{a_1, ..., a_m\}$ and the vectors $\{b_1, ..., b_p\}$ let us notice that $a_1, ..., a_m, b_1, ..., b_p$ are orthogonal to each other which means that they are also linear independent, let's denote the dimension of $a_i s$ and $b_i s$ as d, because we have at least m+p independent vectors we can say that $m + p \leq d$, we also know by a fundamental theorem in linear algebra that we can add d - (m+p) columns to this matrix such that all of the columns will be linearly independent (complete the linear independent set to a basis of the space), so will get a matrix of dimensions $(d \times d)$ let's denote this matrix S. this matrix is invertible, let us denote the inverse matrix as S^{-1}

now let's define D as a $(d \times d)$ matrix and such that all of the elements of the matrix are zero except the first m elements of the diagonal line which are equal to 1 let's denote A as A = SDS^{-1} notice that the eigenvectors of A are $a_1, ..., a_m, b_1, ..., b_p$ and the co-responding eigenvalues are 1 for $a_1, ..., a_m$ and 0 for $b_1, ..., b_p$ which means that for every i $\in [m]$ we get that $Aa_i = a_i$

and for every i $\in [p]$ $Ab_i = 0$ as defined in the question v_a is a linear combination of $a_1, ..., a_m$ so there are $c_1, ..., c_m$ scalars such that $v_a = c_1 a_1 + ... + c_m a_m$ and v_b is a linear combination of $b_1, ..., b_p$ so there are $d_1, ..., d_p$ scalars such that $v_b = d_1 b_1 + ... + d_p b_p$ so we get $A(v_a + v_b) = Av_a + Av_b = A(c_1 a_1 + ... + c_m a_m) + A(d_1 b_1 + ... + d_p b_p) = (c_1 Aa_1 + ... + c_m Aa_m) + (d_1 Ab_1 + ... + d_p Ab_p) = (c_1 a_1 + ... + c_m a_m) + (d_1 * 0 + ... + d_p * 0) = (c_1 a_1 + ... + c_m a_m) = v_a$ as needed (all of the transitions used basic properties of matrices)

b) let k_a and k_b be the co-responding vectors for v_a and v_b let's notice that as defined in the question for each $i \neq a$ it holds that ${k_a}^T k_i = 0$ and also for each $i \neq b$ it holds that ${k_b}^T k_i = 0$, and that for each $i \in \{a, b\}$ ${k_i}^T k_i = \|k_i\|^2 = 1^2 = 1$ let's define q as $q = \beta(k_a + k_b)$ where β is a big constant we get that ${k_a}^T q = {k_a}^T \beta(k_a + k_b)$ and by the distributive law of the dot product we get ${k_a}^T q = {k_a}^T \beta k_a + {k_a}^T \beta k_b = \beta {k_a}^T k_a + \beta {k_a}^T k_b = \beta \|k_i\|^2 + 0 = \beta \cdot 1 = \beta$ and for the same reasons: ${k_b}^T q = \beta$ also, notice that for each $i \neq a, b$ it holds that ${k_i}^T q = \beta {k_i}^T k_a + \beta {k_i}^T k_b = \beta \cdot 0 + \beta \cdot 0 = 0$ which means ${k_i}^T q = 0$ so we get: $c = \sum_{i \neq a,b} v_i \frac{\exp({k_i}^T q)}{\sum_{i \neq a,b} \exp({k_i}^T q) + \exp({k_a}^T q) + \exp({k_b}^T q)} + v_a \frac{\exp({k_a}^T q)}{\sum_{i \neq a,b} \exp({k_i}^T q) + \exp({k_a}^T q) + \exp({k_b}^T q)} + v_b \frac{\exp({k_b}^T q)}{\sum_{i \neq a,b} \exp({k_i}^T q) + \exp({k_a}^T q) + \exp({k_b}^T q)} = \sum_{i \neq a,b} v_i \frac{\exp(0)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)} + v_a \frac{\exp(\beta)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)} + v_b \frac{\exp(\beta)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)}$ let's notice that the limit of $\frac{\exp(0)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)}$ when beta grows to infinity is 0, and the limit of $\frac{\exp(\beta)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)}$ is $\frac{1}{2}$ when beta grows to infinity. so for a big enough β we can make the expression $c = \sum_{i \neq a,b} v_i \frac{\exp(0)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)} + v_a \frac{\exp(\beta)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)} + v_b \frac{\exp(\beta)}{\sum_{i \neq a,b} \exp(0) + \exp(\beta) + \exp(\beta)}$ to get as close as we want to $\sum_{i \neq a,b} v_i \cdot 0 + v_a \cdot \frac{1}{2} + v_b \cdot \frac{1}{2} = \frac{1}{2}(v_a + v_b)$ so in other words $c \approx \frac{1}{2}(v_a + v_b)$ as needed

1.3

a) let's assign $q = \beta(\mu_a + \mu_b)$ where β is a big constant, let's notice that $k_a \sim \mathcal{N}(\mu_a, \alpha_a I)$ and $k_b \sim \mathcal{N}(\mu_b, \alpha_b I)$ where α_a and α_b is vanishingly small numbers, so we can state that in both k_a and k_b the variance of each coordinate is a vanishingly small number, and that there is independence between different coordinates so because the variance is so small for each coordinate, we can say that $k_a \approx \mu_a$ and $k_b \approx \mu_b$ and so we can say that $q \approx \beta(k_a + k_b)$ and so from the previous section we can state that $c \approx \frac{1}{2}(v_a + v_b)$

b) using $q = \beta(\mu_a + \mu_b)$ from the last section, we will get a value c that has high variance and might not be similar to the value we would have wanted. Since in this case, k_a has a high variance, μ_a is no longer a guaranteed approximation, and because of k_a's high variance, while using the same q we get a value c that for multiple runs can have very different values $\rightarrow$ a degraded quality.

1.4

a) We will choose λ , a very large scalar, such that:
$q_1 = \lambda k_a$
$q_2 = \lambda k_b$
λ is a large scalar to ensure that the softmax operation approximates the average of the two vectors, as hinted in hint 2. The scalar λ ensures that q_1 and q_2 heavily align with k_a and k_b, making these keys dominate in the attention mechanism and hence making c_1 and c_2 close to v_a and v_b respectively.

b) This change affects the query vector that's paying attention to k_a (which is q1), leading to a bit more variation in c1. But for the query vector that's paying attention to k_b (which is q2), it doesn't see much difference because the distribution of key vectors related to k_b hasn't changed. Therefore, the output c2, associated with q2, does not vary significantly. Thus overall, c which is the average of c1 and c2, will have slightly to none change.

2 Question 2

2.1

2.2

2.3

Evaluating on the dev set was 2% successful.
Predicting London as the birthplace for everyone in the set was 5% successful.

2.4

2.5

2.6

Output: Correct: 72.0 out of 500.0: 14.399999999999999

2.7

We start with a model that's already been trained, which means it has learned things before. Specifically, it learned from Wikipedia, so it knows a lot and understands how the English language better. When we adjust this model a bit to suit our needs - finetune it, it still remembers what it learned. That way, it performs better when tested.