1a.

The Formula for Dirichlet Prior Smoothing is given by

P(w / d) = + P(w / C)

Let us consider we have a very large document and |d| tends to infinite.

Before that let’s refactor equation by dividing the numerator and denominator by |d|, we have

P(w / d) = + P(w / C)

Since we have |d| tending to infinity and we know 1/x at x tends to infinity is zero substituting it we get

P(w / d) = + P(w / C)

**P(w / d) =**

In a similar way when tends to infinity, we have the equation after dividing the numerator and denominator with we have

P(w / d) = + P(w / C)

Since we have tending to infinity and we know 1/x at x tends to infinity is zero substituting it we get

P(w / d) = + P(w / C)

**P(w / d) = P(w / C)**

1b.

Katz smoothing applies Good-Turing estimates to the problem of backoff language models.Katz smoothing uses a form of discounting in which the amount of discounting is proportional to that predicted by the Good-Turing estimate. Katz smoothing for higher-order n-grams is defined analogously. Katz smoothing doesn’t consider any background model to estimate the terms rather it uses the previous n-1 gram to generate the n-gram. Since Katz is a generative model, the probability obtained may not be accurate, Jelinek-Mercer is better in this aspect since it uses background model which is better accurate. Ability to use Background Knowledge is an advantage.

2a.

Given document d is d: “the sun rises in the east and sets in the west”

The length of the document is |d| = 11

We know that from Unigram Language Model (w / d) = using this we get

|  |  |
| --- | --- |
| Word | (w / d) |
| a | 0 |
| the | 3/11 |
| from | 0 |
| retrieval | 0 |
| sun | 1/11 |
| rises | 1/11 |
| in | 2/11 |
| BM25 | 0 |
| east | 1/11 |
| sets | 1/11 |
| west | 1/11 |
| and | 1/11 |

The Formula for Dirichlet Prior Smoothing is given by

P(w / d) = + P(w / C)

Using this formula we get

|  |  |  |  |
| --- | --- | --- | --- |
| Word | (w / d) | *P(w / C)* | *P(w /d) for* |
| a | 0 | 0.18 | 0.048 |
| the | 3/11 | 0.17 | 0.2453 |
| from | 0 | 0.13 | 0.0346 |
| retrieval | 0 | 0.02 | 0.0053 |
| sun | 1/11 | 0.05 | 0.08 |
| rises | 1/11 | 0.04 | 0.0773 |
| in | 2/11 | 0.16 | 0.176 |
| BM25 | 0 | 0.01 | 0.00266 |
| east | 1/11 | 0.02 | 0.072 |
| sets | 1/11 | 0.04 | 0.0773 |
| west | 1/11 | 0.02 | 0.072 |
| and | 1/11 | 0.16 | 0.1093 |

2b.

Using the same approach for and we get

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Word | (w / d) | *P(w / C)* | *P(w /d) for* | *P(w /d) for* |
| a | 0 | 0.18 | 0.000163 | 0.1621 |
| the | 3/11 | 0.17 | 0.27263 | 0.1801 |
| from | 0 | 0.13 | 0.00011 | 0.1171 |
| retrieval | 0 | 0.02 | 0.0000181 | 0.0180 |
| sun | 1/11 | 0.05 | 0.09087 | 0.0540 |
| rises | 1/11 | 0.04 | 0.09086 | 0.0450 |
| in | 2/11 | 0.16 | 0.181 | 0.1621 |
| BM25 | 0 | 0.01 | 0.0000090 | 0.00900 |
| east | 1/11 | 0.02 | 0.09084 | 0.02702 |
| sets | 1/11 | 0.04 | 0.09086 | 0.0450 |
| west | 1/11 | 0.02 | 0.09087 | 0.0270 |
| and | 1/11 | 0.16 | 0.09097 | 0.1531 |

In 2(a) is significant compared to d so the Background model has affect on the probabilities and the distribution is normalized, the value of is dominant since d>. In case of 2(b) when = 0.01 here is almost tending to zero and is a small value and hence the resulting probability will be almost equal to I.e P(w / d) . In the second case where =100, here is very large compared to d and it has complete dominance, hence in this case P(w / d) .

2c.

We know that Jelinek-Mercer smoothing is given by the formula

P(w / d) = (1 - λ) + P(w / C)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Word | (w / d) | *P(w / C)* | *P(w /d) for* | *P(w /d) for* | *P(w /d) for 0.9* |
| a | 0 | 0.18 | 0.0018 | 0.09 | 0.162 |
| the | 3/11 | 0.17 | 0.2717 | 0.221 | 0.1802 |
| from | 0 | 0.13 | 0.0013 | 0.0013 | 0.117 |
| retrieval | 0 | 0.02 | 0.0002 | 0.01 | 0.018 |
| sun | 1/11 | 0.05 | 0.0905 | 0.07045 | 0.054 |
| rises | 1/11 | 0.04 | 0.0904 | 0.0654 | 0.045 |
| in | 2/11 | 0.16 | 0.1816 | 0.17090 | 0.1621 |
| BM25 | 0 | 0.01 | 0.0001 | 0.005 | 0.009 |
| east | 1/11 | 0.02 | 0.0902 | 0.0554 | 0.02709 |
| sets | 1/11 | 0.04 | 0.0904 | 0.0654 | 0.0450 |
| west | 1/11 | 0.02 | 0.0902 | 0.0554 | 0.0270 |
| and | 1/11 | 0.16 | 0.0916 | 0.1254 | 0.1530 |

When is low as 0.01 the resultant probability is mostly dominated by the and we have P(w / d) . When is 0.5 the resultant probability is average of the two terms and hence both have equal affect on the resultant probability P(w / d) = 0.5\* + 0.5\*P(w / C). When is as high as 0.9, the resultant probability is completely dominated by P(w /C) and we have P(w / d) . Comparing this with the results in 2(a) and 2(b) we know that = and (1- ) is equivalent to , whenever or is high P(w / d) , and whenever (1- ) or is high P(w / d) .

3a.

The RSJ model is given by the formula log(O(R=1/ Q,D) ),

O(R=1/Q,D) = = , we are ignoring the terms

, , then we will have

O(R=1/Q,D)

Let D consists of words (w1,w2,w3……..,wd) which are independent of each other. Since words are independent we have

P(D / Q,R=1) = P(w1 / Q,R=1)P(w2 / Q,R=1)…………….P(Wd / Q,R=1)

We can write it as a product of words present in the vocabulary and we can iterate over all the words in the vocabulary and then if the word which is present in the vocabulary is not present in the document we can give it a power zero.

P(D / Q,R=1) = where says whether w belongs to document D or not.

We can write the power term as the frequency of the word w in the Document and do this for all the words in the document and hence we get.

P(D / Q,R=1) = , where c(w,D) is the frequency of the word in document D.

In a similar way we can show that

P(D / Q,R=0) =

Now writing down both the terms in the equation above, we get

O(R=1/Q,D) = =

Now we have O(R=1/Q,D) taking log of it to get the RSJ model we get.

Log(O(R=1/Q,D)) = log( ) , we know that log(ab) = log(a) + log(b) and

log( applying these two to the equation we get

Log(O(R=1/Q,D)) =

Given Score(Q,D) = Log(O(R=1/Q,D)) and hence we can say that,

Score(Q , D) =

We require 2\*|V| parameters in a retrieval model. Where |V| is the Document length as well since we are considering only one document D. if |V| has more words, we just require those words which are present in document because c(w,D) will be zero if word is not present in a document. If we know 2\*|V| -1 parameters, we can find the 2\*|V|th parameter.

We know that P(w) = + so if we know 2 of the three terms present there we will have the parameters for the retrieval model. To know the two parameters we require 2\*|V| parameters in total.

3b.

P(w / Q,R=0) = The Probability of word w present in an irrelevant document.

Given that the no of irrelevant documents is a collection of N documents C = {D1,D2……..Dn) and hence

P(w / Q,R=0) = =

Maximum likelihood estimate is given by

P(w / Q,R=0) =

3c.

P(w / Q,R=1) = The Probability of word w present in a relevant document.

Given that the no of relevant documents is just the query, and hence no of relevant documents is 1.

P(w / Q,R=1) = =

Since we are considering Query is the only relevant document, we will have

P(w / Q,R=1) =

3d.

We know that Jelinek-Mercer smoothing is given by the formula

P(w / d) = (1 - ) + P(w / C) , using it we get

P(w / Q,R=1) = (1 - ) + LM is background model.

P(w / Q,R=1) = (1 - ) +

3e.

Score(Q , D) =

From 3b and 3d we have

P(w / Q,R=0) = , P(w / Q,R=1) = (1 - ) + , substituting it we get.

Score(Q , D) = )

Score(Q , D) = )

The retrieval function can capture the TF(which is the green part c(w,D)). It can also capture the relative term frequency which is inside the log. the retrieval function cannot capture the IDF and Document length Normalization.