Information Retrieval (5LN712)

Language Models for Information Retrieval

Ali Basirat

Department of Linguistics and Philology
Uppsala University

May 20, 2020

Table of contents

Introduction

- If a document is likely to generate a query, the document is probably relevant to the query
- A language model M_d is trained for each document d
- For each query q, documents are ranked based on $P(q|M_d)$
- This is the probability of the document model M_d generating the query q

Table of contents

Language Model

- A language model assigns a probability to sequences of words
- The main problem is to calculate the probability

$$P(w_1, \ldots, w_n)$$

- Language models are essential parts of NLP and speech recognition methods
- The automata theory is used to study language models

Language Model

Finite Automata

- An abstract machine that can recognize and generate strings
- As a recognizer, a finite automaton tells us whether a string belongs to a language or not
- As a generator, a finite automaton generates a set of strings that form a language

Language Model

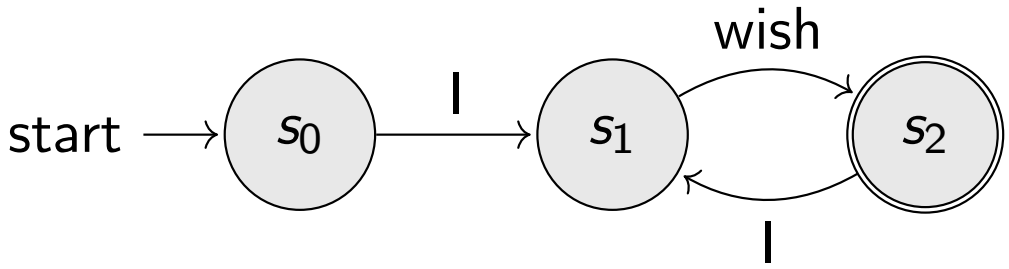
Finite Automata

- A finite automaton consists of
  - A finite set of states including an initial state and a set of accept states
  - A finite set of symbols (terms)
  - A transition function that guides how to move between states based on input symbols

Language Model

Finite Automata

Example



- I wish
- I wish I wish
- I wish I wish I wish
- ...

Language Model

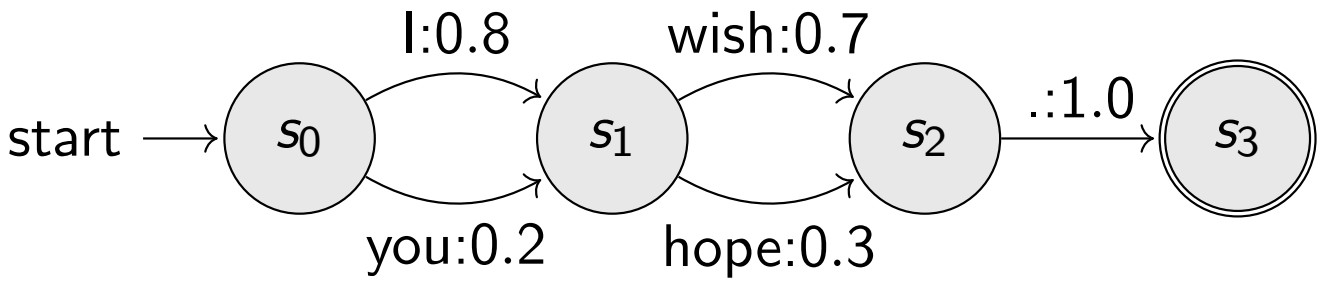
Probabilistic Finite Automata

- The transition between states is according to a probability function
- At each state, the symbols are generated based on a probability distribution

Language Model

Probabilistic Finite Automata

Example



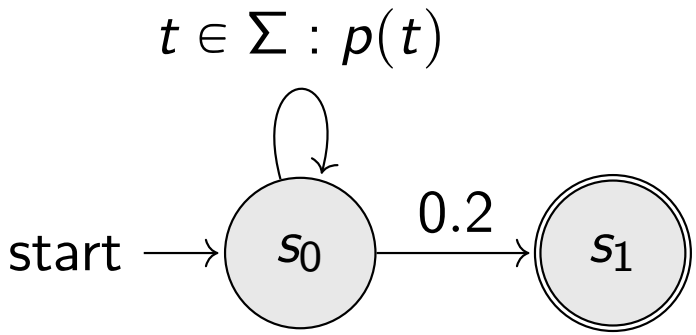
- $P(\text{I wish .}) = 0.8 \times 0.7 \times 1.0 = 0.56$
- $P(\text{you wish .}) = 0.2 \times 0.7 \times 1.0 = 0.14$
- $P(\text{I hope .}) = 0.8 \times 0.3 \times 1.0 = 0.24$
- $P(\text{you hope .}) = 0.2 \times 0.3 \times 1.0 = 0.06$
- $0.56 + 0.14 + 0.24 + 0.06 = 1.0$

Language Model

- A language model is a probabilistic finite automaton
- It defines a probability distribution over all strings of a language
- This means, the sum of the probabilities of the strings made by the set of language words is one

$$\sum_{s \in \Sigma^*} s = 1$$

Example



- $\Sigma = \{\text{the,a,frog,toad,said,like,that,}\ldots\}$

t	$p(t)$
the	0.2
a	0.1
frog	0.01
toad	0.01
said	0.03
likes	0.02
that	0.04
...	

Example

- The probability of `frog said that toad likes frog` is:

$$\begin{aligned} & 0.01 \times 0.03 \times 0.04 \times 0.01 \times 0.02 \times 0.01 \\ & \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.2 \\ & \approx 0.000000000001573 \end{aligned}$$

- The first row is the probability of seeing each term, called the emission probability
- The second row is the probability of continuing or stopping after each term is produced

Language Model

- The probability values are often very small
- Often only the emission probabilities are considered
- It is because we are usually interested in the likelihood ratio of language models to generate a string

Language Model

Example

Example

term	M_1	M_2
the	0.2	0.15
a	0.1	0.12
frog	0.01	0.0002
toad	0.01	0.0001
said	0.03	0.03
likes	0.02	0.04
that	0.04	0.04
dog	0.005	0.01
cat	0.003	0.015
monkey	0.001	0.002
...	...	...

- s: frog said that toad likes that dog
- $p(s|M_1) = 0.01 \times 0.03 \times 0.04 \times 0.01 \times 0.02 \times 0.04 \times 0.005 = 0.00000000000048$
- $p(s|M_2) = 0.0000000000000384$
- $p(s|M_1) > p(s|M_2)$: M_1 is more likely to generate s

Language Model

Different Types

- The probability of a sequence $s = t_1 t_2 \dots t_n$ is

$$p(t_1 \dots t_n) = P(t_1)P(t_2|t_1)P(t_3|t_1 t_2) \dots P(t_n|t_1 \dots t_{n-1})$$
$$= P(t_1)P(t_2|t_1) \prod_{i=3}^{n} P(t_i|t_1 \dots t_{i-1})$$

- We often estimate the probabilities of short pieces of sequences on which the probabilities are conditioned

Language Model

Different Types

- Unigram language model: the condition parts are completely ignored

$$p(t_1 \dots t_n) = \prod_{i=1}^{n} P(t_i)$$

- Bigram language model: the probabilities are estimated based on the previous term

$$p(t_1 \dots t_n) = P(t_1) \prod_{i=2}^{n} P(t_i | t_{i-1})$$

Language Model

Different Types

- More complex language models can also be imagined
- IR systems (studied in this course) mostly use unigram language models
- Unigram models does not take the structure of a sentence into account
- These models tell us about the probability of the presence of terms
- These models are called "bag-of-words" since they do not take the order of the words into account

Language Model

Multinomial Distributions over Words

- From the point of view of a unigram language model, a document is a bag of words
- Hence, documents follow a multinomial distribution
- The probability of a document d of length d is:

$$P(d) = \frac{L_d!}{\prod_{i=1}^{M} \mathrm{tf}_{t_i,d}!} \prod_{i=1}^{M} P(t_i)^{\mathrm{tf}_{t_i,d}}$$

- Since we consider only the likelihood ratio of document probabilities, the multinomial coefficient can be removed:

$$w(d) = \prod_{i=1}^{M} P(t_i)^{\mathrm{tf}_{t_i,d}}$$

Table of contents

Query Likelihood LM

- A language model is trained for each document in the collection
- For each query q, the relevance of documents is proportional to $P(d|q)$

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}$$

- The denominator $P(q)$ is a constant for all documents and have no effect on the rankings
- the prior $P(d)$ is often set uniformly, hence it can be eliminated too

$$P(d|q) \approx P(q|d)$$

Query Likelihood LM

- The probability $P(q|d)$ measures the extent to which the query q can be generated from the document d
- Documents are ranked based on the probability that the query can be sampled from the document
- If we use a multinomial unigram language model M_d for each document d:

$$P(q|d) \approx P(q|M_d) = \frac{L_q!}{\prod_{i=1}^{M} \text{tf}_{t_i,q}!} \prod_{i=1}^{M} P(t_i|M_d)^{\text{tf}_{t_i,q}}$$

- The multinomial coefficient is ignored since it is a constant for a query

$$P(q|M_d) \propto \prod_{i=1}^{M} P(t_i|M_d)^{\text{tf}_{t_i,q}}$$

Query Likelihood LM

The Retrieval Procedure

- Infer a language model for each document d_i, $i = 1, \ldots, n$
- For each query q, estimate the probability of generating the query given each of the document models, $P(q|M_{d_i})$
- Rank the documents according to the probabilities

Table of contents

The Query Generation Probability

- We know that $P(q|M_d) \propto \prod_{i=1}^{M} P(t_i|M_d)^{\text{tf}_{t_i,q}}$
- Since $P(t_i|M_d)^{\text{tf}_{t_i,q}}$ is equal to one for those terms that do not appear in the query $(\text{tf}_{t_i,q} = 0)$, we have:

$$P(q|M_d) \propto \prod_{t \in V_q} P(t|M_d)^{\text{tf}_{t,q}} = \prod_{t \in q} P(t|M_d)$$

- V_q is the set of terms in q
- The maximum likelihood estimate of each of the term probabilities $P(t|M_d)$ is:

$$\hat{P}_{\text{mle}}(t|M_d) = \frac{\text{tf}_{t,d}}{L_d}$$

The Query Generation Probability

Example

- We have a collection of two documents:
  - d_1 : How the coronavirus took advantage of humanity's essential weakness ?
  - d_2 : The economy is drastically affected by the coronavirus pandemic .
- Let $q =$ `coronavirus advantage` be a query
- $P(q|d_1) \propto (\frac{1}{10})^1 \times (\frac{1}{10})^1 = 0.01$
- $P(q|d_2) \propto (\frac{1}{10})^1 \times (\frac{0}{10})^1 = 0$

The Query Generation Probability

Probability Smoothing

- Terms that do not appear in a document have a zero probability
- The presence of 0 in the product $\prod_{t \in V_q} P(t|M_d)^{\text{tf}_{t,q}}$ is problematic
- The smoothing technique cancels out the effect of these values
- A small probability mass is associated with the unseen words
- This probability mass is smaller than the probability of seeing the word in the entire collection

$$P(t|M_d) \leqslant \frac{\text{cf}_t}{T}$$

The Query Generation Probability

Probability Smoothing

Two smoothing methods:

- The linear interpolation: the term probabilities are estimated from a mixture model of two components, one is built on the document, and the other is built on the collection

$$\hat{P}(t|d) = \lambda \hat{P}_{\text{mle}}(t|M_d) + (1-\lambda)\hat{P}_{\text{mle}}(t|M_c)$$

- Bayesian smoothing: a value proportional to term prior probability is added to the frequency counts

$$\begin{aligned}\hat{P}(t|d) &= \frac{\text{tf}_{t,d} + \alpha \hat{P}(t|M_c)}{L_d + \alpha} \\ &= \frac{L_d \hat{P}(t|M_d) + \alpha \hat{P}(t|M_c)}{L_d + \alpha}\end{aligned}$$

The Query Generation Probability

Probability Smoothing

- We can associate documents with a prior probability $P(d)$
- Using the linear interpolation smoothing, the probability that a document d generates a query q is:

$$P(q|d) \propto P(d) \prod_{t \in V_q} (\lambda P(t|M_d) + (1-\lambda) P(t|M_c))^{\mathrm{tf}_{t,q}}$$

- If we use a Bayesian smoothing, the probability is:

$$P(q|d) \propto P(d) \prod_{t \in V_q} \left(\frac{L_d}{L_d+\alpha} P(t|M_d) + \frac{\alpha}{L_d+\alpha} P(t|M_c)\right)^{\mathrm{tf}_{t,q}}$$

$$= P(d) \prod_{t \in V_q} (\lambda P(t|M_d) + (1-\lambda) P(t|M_c))^{\mathrm{tf}_{t,q}}$$

where $\lambda = \frac{L_d}{L_d+\alpha}$.

The Query Generation Probability

Example

- We have a collection of two documents:
  - d_1 : How the coronavirus took advantage of humanity's essential weakness ?
  - d_2 : The economy is drastically affected by the coronavirus pandemic .
- Let $q = \texttt{coronavirus advantage}$ be a query, $\lambda = \frac{1}{2}$, and $P(d_1) = P(d_2)$
- Since the prior probabilities are equal, we ignore them
- $P(q|d_1) = \frac{1}{2}(\frac{1}{10} + \frac{2}{20})^1 \times \frac{1}{2}(\frac{1}{10} + \frac{1}{20})^1 = \frac{3}{400} = 0.0075$
- $P(q|d_2) = \frac{1}{2}(\frac{1}{10} + \frac{2}{20})^1 \times \frac{1}{2}(\frac{0}{10} + \frac{1}{20})^1 = \frac{1}{400} = 0.0025$
- $P(q|d_1) > P(q|d_2)$

Table of contents

Language Modeling vs Other Approaches in IR

- To look at IR from a LM point of view
- The LM model approach is similar to the tf-idf weighting
- The LM model provides a better weighting scheme than for example tf-idf
- The LM model is more accurate than BIM
- It is difficult to integrate relevance feedback to query likelihood LM models of IR

Table of contents

Extended LM Approaches

Document Likelihood Model

- To reverse the generation direction
- To maximize the document likelihood instead of the query likelihood
- Instead of the probability of a document generating a query, we consider the probability of a query generating a document
- The relevance feedback integrates easier with these models.
- The query is expanded with the relevance feedback terms

Extended LM Approaches

General Risk Minimization

- To measure the statistical distance between a document model M_d and a query model M_q
- We minimize the risk of retrieving a document d as relevant to query q
- This can be done by minimizing the KL divergence between the document model M_d and the query model M_q

$$R(d;q) = KL(M_d||M_q) = \sum_{t \in V} P(t|M_d) \log \frac{P(t|M_d)}{P(t|M_q)}$$

- This approach outperforms both query likelihood and document likelihood models

Extended LM Approaches

Translation Models

- To use translation models for IR
- Translation models address the synonymy in languages
- A translation model can generate query terms that are not in a document
- Query terms that are not in a document are expanded to alternate terms with similar meaning
- This provides for cross-lingual IR
- The synonymy is modelled by a conditional probability between words

$$P(q|M_d) = \prod_{t \in V_q} \sum_{v \in V} P(v|M_d) p(t|v)$$

Table of contents

Summary

- Language modelling can be effectively used for information retrieval
- The query likelihood model
- To maximize the probability of a document generating a query
- A document model is trained for each document
- Documents are ranked according to the probability of generating a query from the document models
- Smoothing is needed because it is likely that some words in a query are not present in a document
- Language modelling is more accurate than other approaches of IR
- The query likelihood model can be extended in different ways

References I

Manning, Christopher D. and Raghavan, Prabhakar and Schütze, Hinrich.
Introduction to Information Retrieval.
Cambridge University Press, 2008.