

# Term-Specific Smoothing for the Language Modeling Approach to Information Retrieval: The Importance of a Query Term

Djoerd Hiemstra  
University of Twente  
SIGIR'02

Presentation by:  
Gabriel Ripoché <[gripoché@uiuc.edu](mailto:gripoché@uiuc.edu)>

CS491CXZ – Information Retrieval  
February 19<sup>th</sup>, 2004

# Presentation map

---

- The language modeling approach to information retrieval
- The importance of smoothing
- Presentation of Hiemstra's paper
- Discussion of Hiemstra's paper
- Questioning current approaches to information retrieval

# The language modeling approach to information retrieval

*Concept*

- Estimate a language model for each document:  $M_d$
- Estimate probability of *generating* a query  $Q$  using a given document model:  $P(Q|M_d)$
- Rank documents by probability of generating  $Q$

- LMs come from research in automatic speech recognition
- Goal is to estimate the probability of generating a phoneme given a LM and a series of previous phonemes
- Based on  $n^{\text{th}}$  order Markov chains:  $P(p_i | p_{i-1}, p_{i-2}, \dots)$
- In NLP, most LMs are n-grams ( $n > 1$ )
- In IR, “bag of words” hypothesis: unigram models

## Why do smoothing?

- *Data sparseness*: It is not because a word is not present in a documents that it is impossible that such word could exist in a similar context

## How does it work?

- The probabilities of seen events are decreased and the remaining “probability mass” is allocated to unseen events (think “just in case” )

# The importance of smoothing

## *Smoothing applied to LM-based IR*

$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n P(T_i | D)$$

**Simple model:** if a term isn't in document, query can't be generated ( $P=0$ )

$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n ((1-\lambda) P(T_i | C) + \lambda P(T_i | D))$$

Weight given  
to smoothing

**Smoothing:** probability of term in collection also taken into account

### Current IR systems

- Statistics-based IR doesn't allow user's specification of importance of words<sup>1</sup> (think '-' and '+' in Altavista Search)
- The user should be able to override the default ranking mechanism

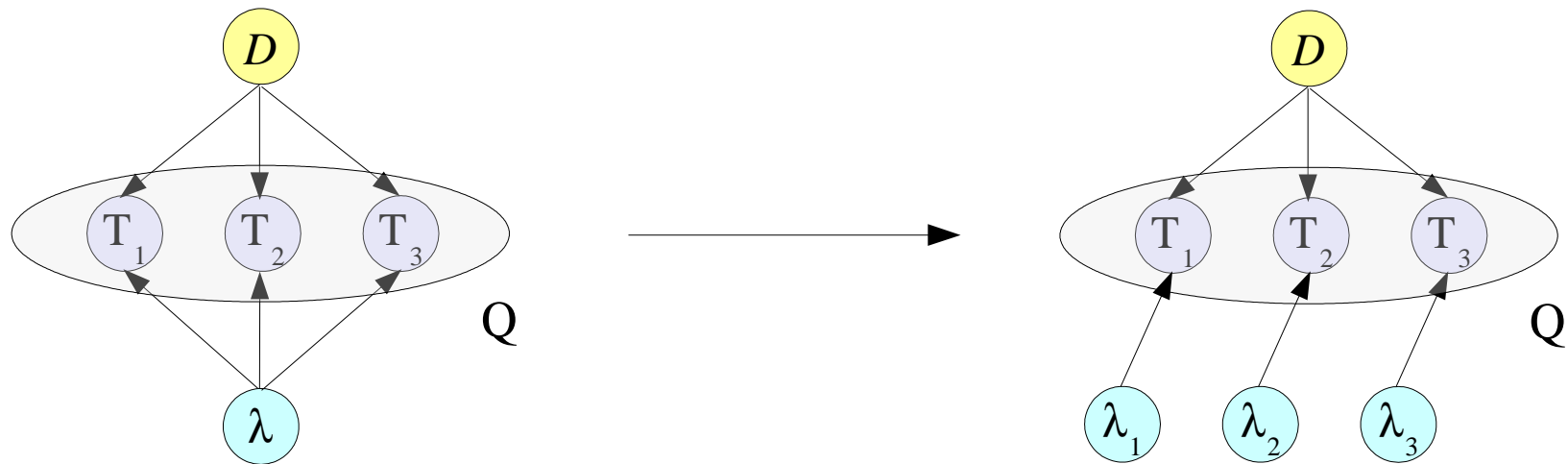
### Objective

- Mathematical model that supports the concept of *query term importance*

---

<sup>1</sup> What about term weighting?

From collection-level smoothing to term-specific smoothing



$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n ((1 - \lambda) P(T_i | C) + \lambda P(T_i | D))$$

(Eq 1)

$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n ((1 - \lambda_i) P(T_i | C) + \lambda_i P(T_i | D))$$

(Eq 2)

# Hiemstra's paper

## Derivation of term importance

$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n (P(T_i | D))$$

Independence between  $T_i$

$$= \prod_{i=1}^n \left( \sum_{i_i \in \{0,1\}} P(T_i, I_i = i_i | D) \right)$$

$I_i \in \{0,1\}$  : importance of  $T_i$   
Sum over values of  $I_i$

$$= \prod_{i=1}^n \left( \sum_{i_i \in \{0,1\}} P(I_i = i_i) P(T_i | I_i = i_i, D) \right)$$

$I_i$  independent of  $M_{\text{document}}$

$$= \prod_{i=1}^n (P(I_i = 0) P(T_i | I_i = 0, D) + P(I_i = 1) P(T_i | I_i = 1, D))$$

$\lambda_i = P(I_i = 1), \quad (1 - \lambda_i) = P(I_i = 0)$   
 $P(T_i | D) = P(T_i | I_i = 1, D)$   
P(important  $T_i$ ) determined  
by  $M_{\text{document}}$  (term salience)  
 $P(T_i | C) = P(T_i | I_i = 0, D)$   
P(unimportant  $T_i$ ) determined  
by  $M_{\text{collection}}$  (no salience)

$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n ((1 - \lambda_i) P(T_i | C) + \lambda_i P(T_i | D))$$

$$P(T_1, \dots, T_n | D) = \prod_{i=1}^n ((1 - \lambda_i) P(T_i | C) + \lambda_i P(T_i | D))$$

$$\lambda_i = 0 \Rightarrow \lambda_i P(T_i | D) = 0$$

### Stop words ('-')

- Ignore query term
- Term not totally ignored since smoothing ( $M_{\text{collection}}$ ) is used

$$\lambda_i = 1 \Rightarrow (1 - \lambda_i) P(T_i | C) = 0$$

### Mandatory words ('+')

- No smoothing from  $M_{\text{collection}}$
- Documents not matching query are assigned 0 probability

$$\forall i, \lambda_i \rightarrow 1$$

### Coordination level ranking

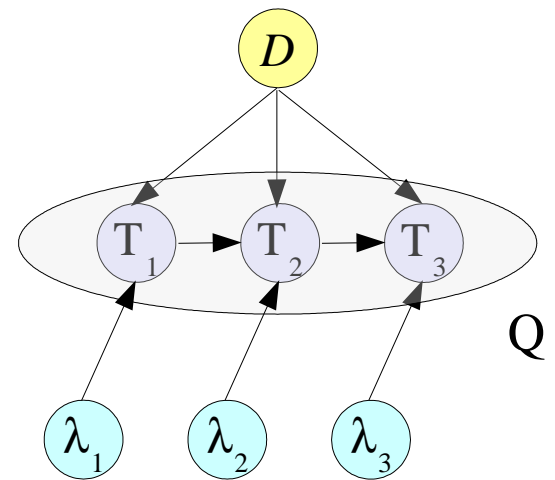
- A document with  $k$  query terms will always rank higher than one with  $k-1$  terms
- Why prove this here? (this is a property of eq. 1)

$$P(T_1, \dots, T_n | D) = (1 - \lambda_1) P(T_1 | C) + \lambda_1 P(T_1 | D) \prod_{i=2}^n ((1 - \lambda_i - \mu_i) P(T_i | C) + \lambda_i P(T_i | D) + \mu_i P(T_i | T_{i-1}, D))$$

(Eq 3)

with  $I_i \in \{0, 1, 2\}$ ,  $\mu_i = P(I_i=2)$ , and  $P(T_i | T_{i-1}, D) = P(T_i | T_{i-1}, I_i=2, D)$

- “last will” of Alfred Nobel  
( $\mu_i > 0$ )
- + “last will” of Alfred Nobel  
( $\mu_i = 1, \lambda_i = 0$ )



### What is the contribution of this paper?

- Adds the notion of *query term importance* (or term specific smoothing) to the language modeling approach to IR

### Questions about the paper

- Is this a valid approach? Or “hijacked” smoothing?
- How is that different from term weighting?
- Why do we want coordination level ranking?
- Is the bi-gram generalization valid and/or useful?

# Questioning current approaches to information retrieval

---

## Why not borrow more from NLP?

- Many approaches only try to do *term matching*.  
What about using *syntax* (bag of words vs. structured text) and *semantics* (exact terms vs. “equivalent” terms)?

## Current IR hypotheses

- What is the validity of coordination level ranking?  
(especially with more semantic approaches)
- What is the difference between LM and “tf.idf” methods?

# References

---

- A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of ACM SIGIR'99*, pp 222-229, 1999.
- **D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *Proceedings of ACM SIGIR'02*, pp 35-41, 2002.**
- C. Manning and H. Schütze. *Foundations of statistical natural language processing*, MIT Press. Cambridge, MA. May 1999.
- J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR'98*, pp 275-281, 1998.
- C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of ACM SIGIR'01*, pp 334-342, 2001.