Passage Retrieval from Genomic Texts: An Experience at TREC 2007

Raymond Wan[1] rwan@kuicr.kyoto-u.ac.jp
Vo Ngoc Anh[2] vo@csse.unimelb.edu.au
Hiroshi Mamitsuka[1] mami@kuicr.kyoto-u.ac.jp

[1] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, 611-0011, Japan
[2] Department of Computer Science and Software Engineering, Faculty of Engineering, University of Melbourne, Victoria, 3010, Australia

Abstract

The Text Retrieval Conference is an annual conference where researchers in information retrieval (IR) compare their systems on specified retrieval tasks. This poster summarizes what the Genomics Track of TREC is, its task for 2007, and a brief account of the work jointly done by Kyoto University and the University of Melbourne this year. Further details about our method can be found in our "Notebook Paper".

1. TREC Genomics Track 2007

About TREC (`http://trec.nist.gov/`):

- Commenced in 1992 by the National Institute of Standards and Technology (NIST).
- A workshop series that allows IR systems to be evaluated on realistic test collections and uniform scoring methodologies.
- Separated into various tracks (Blog, Enterprise, Genomics, Legal, Million Query, QA, and Spam in 2007).

Genomics Track 2007:

- Fifth and final year for the Genomics Track.
- Like 2006, the single task was passage retrieval from a biomedical text collection (full-text).
- Collection contains 162,259 articles from 49 genomics related journals by Highwire Press.
- Articles in HTML format and divided into paragraphs using <p> tags.
- Definition of a passage: A sequence of words that does not include any paragraph tags.

Evaluation for Genomics Track 2007:

- 36 queries such as "Which [PATHWAYS] are mediated by CD44?" (Query #221).
- Each system returns a ranked list of passages for each query.
- Passages are pooled and evaluated by human judges for relevance.
- Effectiveness of each system measured in terms of mean average precision (MAP).

2. Method Overview

Our method consists of two parts:

1. Identify paragraphs relevant to the query using the vector space model (VSM).
2. Apply a probabilistic word-based aspect model to isolate sections of texts.

2.1 Vector Space Model

We employed an impact-based variant of the VSM:

- Each paragraph or query is represented as an n-dimensional vector (n is the number of distinct words).
- The similarity between a query and a paragraph is reported as an integer instead of a floating point value.
- The similarity is calculated using the scalar product.

2.2 Aspect Model

The aspect model has been used in IR to associate words to documents. We modify the method as follows:

- We employ one-mode factor analysis where words are associated with each other.
- We make use of probabilistic latent semantic analysis and the Expectation-Maximization algorithm.

The aim is to start from a matrix of word co-occurrence counts and end up with a matrix of scores through the use of k clusters or latent states.

Parameters tested:

- $k = 100$ clusters
- Stopping condition of 50 iterations or a change in maximum likelihood of less than 0.0001.

3. System Overview

Our two models are implemented as a paragraph-level retrieval system and a passage extraction system. A final post-processing step is used for ranking the passages. (See Figure 1.)

3.1 Scoring Passages

Score derivation creates a matrix of scores of size m by n, where m is the number of unique words in the query and n is the number of unique words in the collection. Scoring then proceeds as follows.

1. Each word j in the paragraph is scored against every word i of the query q.
2. If a paragraph word matches the query word exactly, then a score of c_{max} is added to the paragraph word, where c_{max} is the maximum score in the matrix.
3. Otherwise, the matrix is consulted.

So, for each $j \in d$, the score $s(q, d, j)$ of the paragraph d against the query q with respect to j is:

$$s(q,d,j) = \frac{|D|}{f_j} \times (1 + \log f_{d,j}) \sum_{i \in q} \bar{c}(i,j). \quad (1)$$

The inner summation for "Method 3" is given as:

$$\bar{c}(i,j) = \begin{cases} \ln(1 + \frac{|D|}{f_i}) \times c_{\text{max}} & \text{if } i = j \\ \ln(1 + \frac{|D|}{f_i}) \times c(i,j) & \text{otherwise} \end{cases} \quad (2)$$

In these formulas,

- $f_{d,j}$ is the frequency of j in paragraph d
- $|D|$ is the number of paragraphs
- f_i is the frequency of a query word in the collection
- f_i is the frequency of a paragraph word in the collection

4. Results

We report on two submitted runs to TREC and one additional set of runs:

ID	VSM results	VSM:AM	
`kyoto1`	1000	100	0
`kyoto2`	5000	0	100
`Method-3`	*	*	*

The performance of our submitted runs and the median are:

ID	Document	Aspect	Passage	Passage2
`kyoto1`	0.1892	0.1208	0.0474	0.0209
`kyoto2`	0.1191	0.0302	0.0235	0.0054
Median	0.1897	0.1311	0.0565	0.0377

`Method-3` is a set of runs which incorporates some corrections to programming bugs.

- Graphs of these results against the median (grey horizontal lines) are shown below.
- The dotted, dashed, and solid lines represent 1,000, 5,000, and 10,000 results from the VSM.
- Performance is optimal when VSM alone is used for ranking.
- MAPs below the median, except for our Passage2 scores which are slightly above.

In the future, we plan to further investigate the effect of the parameters in our system.

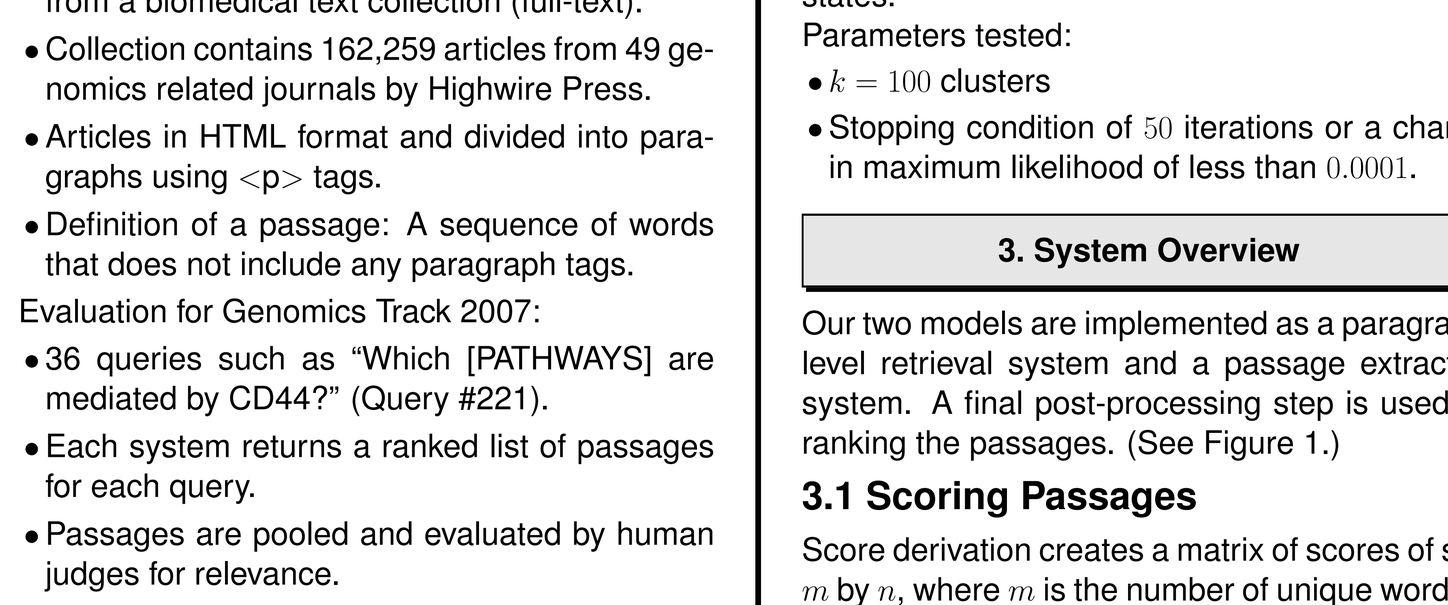

Figure 1: System overview.