Combining Vector-Space and Word-based Aspect Models for Passage Retrieval

Raymond Wan[*] Vo Ngoc Anh[†]
Ichigaku Takigawa[*] Hiroshi Mamitsuka[*]

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

Abstract

This report summarizes the work done at Kyoto University and the University of Melbourne for the TREC 2006 Genomics Track. The single task for this year was to retrieve passages from a biomedical document collection. We devised a system made of two parts to deal with this problem. The first part was an existing IR system based on the vector-space model. The second part was a newly developed probabilistic word-based aspect model for identifying passages within relevant documents (or paragraphs).

1 Introduction

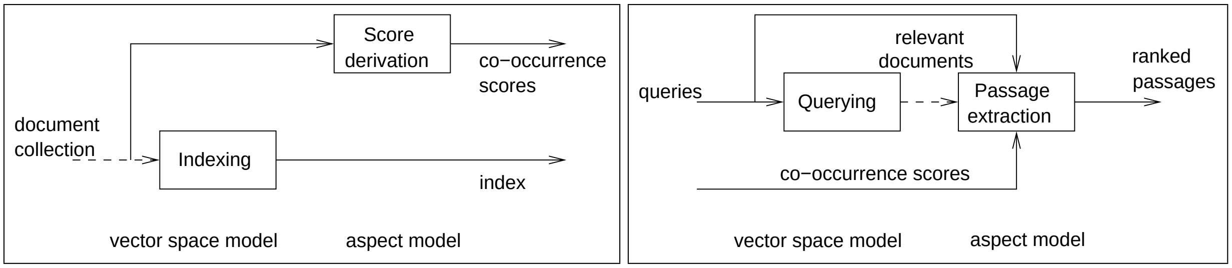
Kyoto University and the University of Melbourne participated together for the TREC 2006 Genomics Track. The task was to retrieve passages from a full-text HTML collection of biomedical journals. A "passage" was defined as a sequence of words which do not cross any paragraph boundaries.

We developed a system comprised of two main parts for passage retrieval. The first is an information retrieval (IR) system which constructs an index for the document collection and returns results according to given queries. The IR system uses a vector-space model (VSM).

The second part employs a probabilistic word-based aspect model (AM) to score a query against a passage by including both exact and inexact word matches. The following example exemplifies the motivation for our work on this part. If word w_1 occurs often with w_2 and w_2 is usually found near w_3, then any two of these words should contribute some positive value to the overall passage score. However, of the three possible scores, the score of w_1 and w_3 should be the lowest. Thus, we assign a score to two words even if they do not match exactly but are usually found within the same paragraph. A preliminary "training" phase is required to derive these scores. In the absence of a separate training set, we used the test collection itself for this purpose.

These two parts of our system are illustrated in Figure 1 and separated between off-line processing Figure 1(a) and on-line processing Figure 1(b).

This report is structured as follows. A description of our system is provided in the next section, with particular emphasis on our use of the aspect model since it is new. Then, in Section 3, we give information on our three officially submitted runs. Ten additional runs were performed just prior to and after TREC in order to compare with the submitted results. These runs are reported next in Section 4. Finally, we summarize our findings in Section 5.



(a) Off-line processing includes indexing and score derivation.

(b) On-line processing includes querying and passage extraction.

Figure 1: Our system includes a vector-space model and an aspect model, which are both required for off-line pre-processing and on-line querying. The purpose of the dashed lines are described later in Section 3.

2 System Description

A description of each part of our system follows. Before beginning with the vector-space model, some points relevant to the entire system (both the vector-space model and the aspect model) are covered.

2.1 System-wide Features

Since certain biological terms appear in various combinations of mixed upper/lower case and hyphenation, we applied biological term normalization with other techniques commonly found in information retrieval systems.

First, we restrict words to contain either alphabetic characters or digits, but not a mix of both. Punctuation characters such as hyphens are considered to be non-word characters by our system. As a result, both “Nurr-77” and “Nurr77” are separated into the two words “Nurr” and “77”. Furthermore, every part of our system makes use of case-folding and stemming using the Lovins algorithm [Lovins, 1968]. All of these transformations are performed on both the query terms and the document collection during both index creation and querying.

A second feature which we call synonym expansion was applied only to query terms. Synonym expansion combines existing information in the query and several external databases to derive lists of words which are similar to the query term. Each query was provided as two parts: a gene and a biological function[1]. First, if an alternate term is given in parentheses, then it is assumed to be a synonym. For example, the gene for query #162 was given as “APC (adenomatous polyposis coli)”. Then, additional synonyms were obtained by looking up certain on-line databases.

If the term is a gene then it was first expanded using the Biomedical abbreviation server[2] [Chang et al., 2002]. The first abbreviation which was an exact match and scored as “Excellent” was returned. In addition, gene names were also expanded using Entrez Gene [Maglott et al., 2005] from the NCBI web site[3]. If multiple candidates were available, the first one was selected. As for biological functions, they were all expanded using the Medical Subject Headings (MeSH terms) [Nelson et al., 2004], also from the NCBI web site in the same way.

Since synonym expansion relied on multiple sources, duplicates in the enlarged query were removed. Synonym expansion can increase the number of words in each query greatly, depending on the query and the number of synonyms found.

[1] During the conference, we realized that this extra information was *not* part of the queries.

[2] `http://abbreviation.stanford.edu/`

[3] `http://www.ncbi.nlm.nih.gov/`

2.2 Vector-space Model

The vector-space model (VSM) is employed in the first stage of our retrieval process. The actual version of VSM used is the impact-based ranking approach. Like in the traditional VSM, the approach represents a text item (where an item can be a document, a paragraph, or a query) by a vector in n-dimension space, where n is the number of distinct terms in the collection.

There are, however, some differences between impact ranking and traditional VSM. First, in the impact ranking approach, all vector coordinates are integers between 1 and 8 (as oppose to floating point values in traditional VSM). Second, the similarity score between two vectors is now calculated as the inner vector product, but not as the cosine of the angle between them, as in conventional practice. The motivations behind these differences as well as the details of the retrieval approach, are described in Anh and Moffat [2005].

2.3 Deriving Scores

The aspect model (also, latent semantic analysis) has been proposed by others to associate words to documents [Hofmann et al., 1998]. In particular, the aspect model maps words and documents to a k-dimensional space using the singular value decomposition (SVD) of co-occurrence tables. By selecting k such that it is less than the number of words or documents, we end up with both words and documents being related to each other through the k latent states, or clusters. Probabilistic latent semantic analysis (PLSA) [Hofmann, 2001] adds a probabilistic model to earlier work by employing an iterative approach using the Expectation-Maximization (EM) algorithm [Dempster et al., 1977].

In these earlier works, the starting point was a co-occurrence table of documents against words. If m denotes the number of documents and n refers to the number of unique words in the collection, then this implies an m by n matrix. In our work, we modify it so that we have an n by n table. This resembles earlier work in the field of document clustering [Borko and Bernick, 1962], except that we retain the methodology employed by Hofmann. So, the score between two different words w_x and w_y is the sum of their scores across all k clusters, where $k \ll n$:

$$p(w_x, w_y) = \sum_{z \in Z} p(w_x|z)p(w_y|z)p(z), \tag{1}$$

In this formula, Z is the set of clusters. The parameters of this aspect model can be estimated using the EM algorithm by iterating between the following E-step and M-step:

E-step:

$$p(z|w_x, w_y) = \frac{p(w_x|z)p(w_y|z)p(z)}{\sum_{z' \in Z} p(w_x|z')p(w_y|z')p(z')} \tag{2}$$

M-step:

$$p(w_x|z) = p(w_y|z) = \sum_{w_y \in W} n(w_x, w_y)p(z|w_x, w_y) \tag{3}$$

$$p(z) = \sum_{w_x \in W} \sum_{w_y \in W} n(w_x, w_y)p(z|w_x, w_y), \tag{4}$$

where W is the set of unique words in the document collection and $n(w_x, w_y)$ is the number of co-occurrences of w_x and w_y within a paragraph across the entire collection. Initial values are generated at random using a uniform distribution.

The output from the word-based aspect model (AM) is a set of scores $p(w_x, w_y)$ such that $p(w_x, w_y) = p(w_y, w_x)$, where $w_x \neq w_y$. For the remainder of this report, we denote these scores as co-occurrence scores, as opposed to co-occurrence counts ($n(w_x, w_y)$). The score of a word with itself was purposely excluded from our aspect model so that it could be defined as some constant, as explained below.

2.4 Passage Extraction

Since the document collection is in HTML format, HTML "p" tags were used to separate each document into paragraphs. However, rules were needed to further divide paragraphs into sections. In our implementation, punctuation marks indicated the boundary between two sections.

Passages were formed from one or more consecutive sections and each was scored against a query through the pair-wise comparison of words. We denote a passage of s words as r_1, ..., r_s and a query of t words as q_1, ..., q_t. The score between two words r_i and q_j (for $1 \leq i \leq s$ and $1 \leq j \leq t$) is defined as:

$$\text{score}(r_i, q_j) = \begin{cases} m & \text{if } r_i = q_j \\ \overline{m} & \text{if } r_i \neq q_j \ . \end{cases} \tag{5}$$

When two words match exactly, then a constant score of m is contributed to the final score for that passage. In the case of a non-match, the score for that passage still increases, but by $\overline{m}$. If the co-occurrence scores obtained through the score derivation step of the aspect model is used, then $\overline{m} = p(r_i, q_j)$. Thus, m is constant, while $\overline{m}$ varies depending on the co-occurrence scores of the two words. Several things are worth noting with respect to Equation (5). As m increases in size compared to $p(r_i, q_j)$ the effectiveness of the co-occurrence scores diminishes. If $\overline{m} = 0$, then the score for the passage is essentially a count of the number of query terms that appear in the passage.

In our experiments, we investigated several values for m and $\overline{m}$. For m, we evaluated three values ranging from $m = p_{\text{max}}$ to $m = 1$, where p_{max} is the maximum score across all word-pairs. Based on how the scores are derived from probabilities, it is obvious that $p_{\text{max}} \ll 1$. As for $\overline{m}$, we also considered the case of $\overline{m} = 0$.

The formula to score an entire passage r against the query q is:

$$\text{score}(r, q) = \frac{\sum_i^s \alpha_{w_i} \times (\sum_i^s \sum_j^t (\text{score}(r_i, q_j)))}{s \times t} \ . \tag{6}$$

As Equation (6) shows, we adjusted the scoring mechanism so that the length of the query and the passage under consideration are taken into account. Moreover, a parameter α_{w_i} is calculated on a paragraph-by-paragraph basis as follows:

$$\alpha_{w_i} = \frac{\text{frequency of } w_i \text{ in paragraph P}}{\text{words in paragraph P}} \times \left(1 - \log \frac{\text{number of documents}}{\text{document frequency of } w_i}\right) \tag{7}$$

$$= \frac{f_{P,w_i}}{|P|} \times \left(1 - \log \frac{|D|}{f_{D,w_i}}\right) . \tag{8}$$

Our motivation for this definition of α_{w_i} was to use the passage and query lengths, the word frequency in the current paragraph, and the IDF factor to affect the scoring mechanism. Furtherwork is necessary in order to determine the merits of the terms in Equation (8). As a first step, a few additional runs were done after TREC.

Run	Indexing level	IR synonym	Documents	Equation (5) m	Weight (VSM:PE)
PARAGRAPH	paragraph	yes	500	1	$\frac{1}{2} : \frac{1}{2}$
DOCUMENT-1	document	yes	1,000	$2 \times p_{\max}$	0 : 1
DOCUMENT-2	document	no	1,000	$2 \times p_{\max}$	0 : 1

Table 1: Submitted runs to TREC.

3 Submissions

Three official runs were submitted to TREC. Ten additional ones were executed prior to and after TREC which were based on the submitted run which performed the best. Details about these additional runs are deferred to the next section.

We applied two types of indexing with the vector-space model for the submitted runs: document-level and paragraph-level indexing. For the document-level, each relevant article is expanded into its constituent paragraphs, prior to passage extraction. This is indicated as a dashed line in Figure 1(b). The VSM score attributed to each paragraph is equal to the score of the document from which it is taken. For paragraph-level indexing, the constituent paragraphs of every article are fed into the VSM for indexing, as shown by the dashed line in Figure 1(a). The output from the VSM to the passage extraction (PE) system is a set of relevant paragraphs. Regardless of the indexing level used, the PE system of the aspect model obtains a set of paragraphs and outputs a list of relevant passages. The score of each passage for final ranking is determined by combining the score given by the VSM and the PE.

As part of the Genomics Track this year, the collection of 162,259 documents included the set of 12,641,127 legal spans (see Hersh et al. [2006] for further details). A legal span is a section of text which excludes any HTML "p" tags. Thus, a legal span is equivalent to the paragraphs described above.

Score derivation by the aspect model was fixed so that the same set of scores is used for all of our runs. The space required by score derivation is $O(kn^2 + kn + k)$ where n is the number of unique words and k is the number of clusters. In order to reduce memory requirements, we pre-selected words based on their document frequency. In the 162,259 documents, there were 1,299,308 unique words using our definition of a "word" from Section 2.1. By choosing words that appeared in at least 1% of the document collection, the lexicon size was reduced to 13,895 unique words. Co-occurrence scores were determined using $k = 128$ clusters and the EM algorithm iterated until the maximum likelihood did not change by more than 1%. The maximum score across all word pairs ($p_{\max}$) was 0.00102.

The maximal number of sections per paragraph was limited to 1,000 due to time constraints. Such paragraphs were usually found as part of bibliographies in articles due to the higher concentration of punctuation marks.

The main characteristics that differed between the official runs are given below and summarized in Table 1.

PARAGRAPH (kyoto1) The document collection was indexed at the paragraph-level by the IR system and the top 500 results (paragraphs) for each query were given to PE. When a query word is equal to a passage word, $m = 1$ in Equation (5). Otherwise, $\overline{m}$ is equal to the co-occurrence scores. Since the final list of results has 1,000 results per query, multiple passages are selected from a single paragraph. The ranking of the results is determined by giving equal weight to the vector-space model and the aspect model.

DOCUMENT-1 (kyoto20) A document-level run using the top 1,000 documents. The paragraphs of every document in the top 1,000 were considered by the aspect model. When a query word is equal to a

Run	Document	Passage	Aspect
PARAGRAPH	0.2248	0.0248	0.1217
DOCUMENT-1	0.1231	0.0075	0.0610
DOCUMENT-2	0.1297	0.0071	0.0692

Table 2: Mean average precision for the three submitted runs.

Phase	Real time	User time
Vector-space model		
Index construction	25	25
Querying	< 1	< 1
Aspect model		
Deriving scores	244	61
Passage extraction	651	632

Table 3: Real and user time (in minutes) for PARAGRAPH with respect to executing the vector-space and aspect models. The computer architecture used differed between the two models; see the text for further details.

passage word (a match), a score of $m = 2 \times p_{\text{max}}$ was used for Equation (5). Ranking was done using only the scores provided by the aspect model.

DOCUMENT-2 (kyoto2) Identical to DOCUMENT-1 except that synonyms were expanded only by the aspect model and not the VSM. This is the only run where the IR system did not make use of synonym expansion.

The mean average precision results from our submission are shown in Table 2. Of the three submitted runs, PARAGRAPH performed the best. This led us to believe that document-level indexing does not perform well for this task and that more granularity is required in the form of paragraph-level indexing.

The running time of PARAGRAPH is given in Table 3 as both real and user time in minutes. Both "Querying" and "Passage extraction" refer to the time required to process all 28 queries. Two different architectures were used for our experiments. The VSM was executed on a 3.06 GHz Intel Xeon with 8 GB RAM, while score derivation and passage extraction were run on a 3.6 GHz Intel Xeon (dual processor) with 8 GB RAM and 8 MB cache.

Since this use of the aspect model is new and generally untested, its execution time is noticeably high. It is expected that a more careful implementation can reduce the running time significantly. The difference between the real and user time for the score derivation phase of the aspect model is due in part to excessive disk swapping because of its high memory requirements.

4 Additional Runs

The performance of PARAGRAPH among our three submitted runs motivated us to concentrate our investigation on this run. In particular, we modified a parameter at a time to help determine which parameters have a noticeable effect on our system's performance. Overall, ten additional runs were performed. A description of each run is given next with respect to PARAGRAPH. In all of these additional runs, paragraph-level indexing was used with synonym expansion. This information is summarized in Table 4.

PARAGRAPH As our reference run using paragraph-level indexing and synonym expansion, 500 paragraphs were taken by the PE system. Only paragraphs with 1,000 sections were considered. In

Run	Documents	Passage Limit	Equation (5)		Equation (6)	Weight (VSM:PE)
			m	$\overline{m}$		
PARAGRAPH	500	1,000	1	$p(r_i, q_j)$	Yes	$\frac{1}{2}:\frac{1}{2}$
AM_1,000	1,000	1,000	1	$p(r_i, q_j)$	Yes	$\frac{1}{2}:\frac{1}{2}$
NO_AM_500	500	N/A	N/A	N/A	N/A	N/A
NO_AM_1,000	1,000	N/A	N/A	N/A	N/A	N/A
NO_SECTION_LIMIT	500	∞	1	$p(r_i, q_j)$	Yes	$\frac{1}{2}:\frac{1}{2}$
MATCH_PMAX	500	1,000	p_{max}	$p(r_i, q_j)$	Yes	$\frac{1}{2}:\frac{1}{2}$
MATCH_2PMAX	500	1,000	$2 \times p_{\text{max}}$	$p(r_i, q_j)$	Yes	$\frac{1}{2}:\frac{1}{2}$
NON_MATCH_0	500	1,000	1	0	Yes	$\frac{1}{2}:\frac{1}{2}$
NO_ALPHA	500	1,000	1	$p(r_i, q_j)$	No	$\frac{1}{2}:\frac{1}{2}$
RANK_WEIGHT_VSM	500	1,000	1	$p(r_i, q_j)$	Yes	$1:0$
RANK_WEIGHT_PE	500	1,000	1	$p(r_i, q_j)$	Yes	$0:1$

Table 4: Additional runs after the release of relevance judgements. In all cases, paragraph-level indexing and synonym expansion were employed, just like PARAGRAPH.

Equation (5), $m = 1$ and $\overline{m} = p(r_i, q_j)$. Equation (6) was used exactly as shown and equal weight was given to both the VSM and the PE scores. This information is summarized in the first rows of Table 1 and Table 4.

AM_1,000 The top 1,000 paragraphs are used by the aspect model.

NO_AM_500 The aspect model was not used. So, each paragraph output through the querying process became a passage. Since there is one passage for each paragraph, the number of passages returned for each query is only 500.

NO_AM_1,000 Similar to NO_AM_500, except that 1,000 paragraphs were output.

NO_SECTION_LIMIT There was no limit on the number of sections allowed in a paragraph.

MATCH_PMAX For exact matches, $m = p_{\text{max}} = 0.00102$.

MATCH_2PMAX For exact matches, $m = 2 \times p_{\text{max}}$.

NON_MATCH_0 When a query term and a passage term does not match, no score is added to the passage score. That is, $\overline{m} = 0$.

NO_ALPHA The first term in the numerator of Equation (6) is removed. In other words, $\sum_i^s \alpha_{w_i} = 1$.

RANK_WEIGHT_VSM The final ranking is determined entirely by the VSM scores.

RANK_WEIGHT_PE The final ranking is determined entirely by the aspect model scores.

The mean average precision results from these runs are given in Table 5. For each type of MAP, our best score is shown in bold font. As these results show, there was no single run which gave the best results for all three measures.

These additional runs can be summarized as follows, with respect to our baseline, PARAGRAPH:

- As expected, by increasing the number of paragraphs given to the PE system from 500 to 1,000, effectiveness improves regardless of whether or not the aspect model is used (see the first four rows of Table 5).

Run	Document	Passage	Aspect
PARAGRAPH	0.2248	0.0248	0.1217
AM_1,000	0.2369	0.0258	0.1235
NO_AM_500	0.2203	0.0103	0.1232
NO_AM_1,000	**0.2372**	0.0117	**0.1246**
NO_SECTION_LIMIT	0.2246	0.0231	0.1204
NON_MATCH_0	0.2231	0.0244	0.1210
MATCH_PMAX	0.1459	0.0083	0.0911
MATCH_2PMAX	0.1558	0.0091	0.0955
NO_ALPHA	0.1497	0.0215	0.0386
RANK_WEIGHT_VSM	0.1744	0.0131	0.0348
RANK_WEIGHT_PE	0.2067	**0.0261**	0.1081

Table 5: Mean average precision for the ten additional runs.

- Removing the restriction that only paragraphs with 1,000 sections or less are processed does not yield significant results (NO_SECTION_LIMIT), despite the fact that some results in the gold standard are found in bibliographies. We believe the reason for this is our system's inability to properly find these relevant passages in bibliographies.

- Setting $\overline{m}$ to 0 reduces the effectiveness of our system slightly for all 3 measures (NON_MATCH_0). In other words, use of the co-occurrence scores ($p(r_i, q_j)$) improves retrieval. This is encouraging since it means the word-based aspect model is helpful in ranking documents and locating passages.

- In contrast, reducing m to a value similar to the maximum co-occurrence score is detrimental to all three measures (MATCH_PMAX and MATCH_2PMAX). Unlike the NON_MATCH_0 run where the difference between m and $\overline{m}$ increases, in these two runs, the difference decreases. So, we hypothesize that the difference between m and $\overline{m}$ should be large, but as NON_MATCH_0 shows, $\overline{m}$ should not be 0.

- Omitting the first term of the numerator of Equation (6) reduces all three measures, but passage retrieval is affected to a lesser degree (NO_ALPHA). Thus, a term which makes use of the IDF factor and paragraph word frequency is useful. However, this run has not proven that this form of the equation is ideal.

- Finally, determining the final ranking by either the VSM or PE degrades performance, except when only the PE system is used. In this case, passage retrieval is the best out of all our runs.

In general, paragraph-level indexing is better than document-level indexing (see Table 2). For paragraph-level indexing, document retrieval and aspect retrieval were best when no passage extraction mechanism was employed (NO_AM_1,000). However, to attain better passage retrieval scores over our reference run (PARAGRAPH), we can simply use the scores from the aspect model (RANK_WEIGHT_PE). An insufficient number of runs were made to extrapolate the effect from combining factors behind the runs. For example, while it is expected that RANK_WEIGHT_PE can be improved by simply using 1,000 paragraphs instead of 500, this has not been demonstrated.

5 Summary

In this report, we have described our contribution to the TREC 2006 Genomics Track. The main task was passage retrieval and our solution was a system made of an IR system based on the VSM and a newly developed word-based aspect model. We initially submitted three runs and then performed ten additional ones to further evaluate our system. In general, the use of co-occurrence scores gives slight improvements for all three measures (see the NON_MATCH_0 run in Table 4), which is encouraging.

However, as our system performed within the bottom half of all systems which submitted runs to the Genomics Track this year, much work is expected. Various parameters and equations need to be further evaluated and the results in Section 4 is only the beginning. The parameters for the aspect model have not been thoroughly examined. In particular, while the reduction of the vocabulary to 13,895 words is necessary due to space limitations, its overall effect is unknown. Moreover, the importance of the number of clusters ($k = 128$) has not yet been properly assessed.

More importantly, our system is actually comprised of three parts: an IR system which returns a set of relevant documents (or paragraphs), an aspect model which derives co-occurrence scores, and a passage extraction system which locates potential passages and scores them. Problems in even one part of this entire system could be problematic. For example, even though our work focusses on the coupling of an information retrieval system with a word-based aspect model, ineffectiveness in the passage extraction system in isolating passages for scoring could have a significant effect on our system's performance. All of these aspects need to be considered as part of our future work.

Acknowledgements: This work was supported in part by the Bioinformatics Education Program "Education and Research Organization for Genome Information Science" and the Kyoto University 21st Century COE Program "Knowledge Information Infrastructure for Genome Science" with support from MEXT (Ministry of Education, Culture, Sports, Science and Technology), Japan. The second author was supported by the Australian Research Council's Center for Perceptive and Intelligent Machines in Complex Environments.

References

V. N. Anh and A. Moffat. Simplified similarity scoring using term ranks. In *Proc. 28th ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 226–233, 2005.

H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM*, 10(2):151–162, 1962.

J. T. Chang, H. Schütze, and R. B. Altman. Creating an online dictionary of abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 9(6):612–620, November-December 2002.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

W. Hersh, A. M. Cohen, P. Roberts, and H. K. Rekapalli. TREC 2006 genomics track overview. In *Proc. 15th Text Retrieval Conference (TREC 2006)*, November 2006.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1–2):177–196, January–February 2001.

T. Hofmann, J. Puzicha, and M. I. Jordan. Learning from dyadic data. pages 466–472, 1998.

J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11: 22–31, 1968.

D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova. Entrez gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33:D54–D58, January 2005.

J. S. Nelson, M. Schopen, A. G. Savage, J.-L. Schulman, and N. Arluk. The MeSH translation maintenance system: Structure, interface design, and implementation. In *Proc. 11th World Congress on Medical Informatics*, pages 67–69, 2004.