

Report on the TREC 2006 Genomics Experiment

Samir Abdou, Jacques Savoy

Computer Science Department, University of Neuchatel
Rue Emile-Argand, 11, CH-2009 Neuchatel (Switzerland)
{Samir.Abdou, Jacques.Savoy}@unine.ch

ABSTRACT

This paper describes our participation in the TREC 2006 Genomics evaluation campaign. In an effort to find text passages that will meet user requests, we propose and evaluate a new approach to the generation of orthographic variants of search terms (mainly genomic names in our case). We also evaluate the retrieval effectiveness of both the Okapi (BM25) model and the I(n)B2 probabilistic model derived from the *Divergence from Randomness* paradigm. In our experiments, we find that in terms of mean average precision the latter model performs clearly better than the Okapi model (with a relative difference of 50%). Moreover when comparing a 5-gram indexing approach to word-based indexing schemes, the mean average precision decreases by about 10% when using the n -gram indexing scheme. Additionally, including the article's title in all passages generated from a given article does not improve retrieval effectiveness. Finally, the generation of passages delimited by HTML tags was not a success. The performance achieved was in fact rather poor, suggesting that there were too many sentences within such text passages.

1. INTRODUCTION

The biomedical domain presents the information retrieval (IR) community with a number of challenging problems. The main objective of the three last Genomics campaigns [1] was thus to retrieve bibliographic references (composed mainly of title, author names and abstract) from a large subset of the MEDLINE repository, in response to real user information needs. The main goal this year is to retrieve text fragments or passages rather than the integral scientific articles available in the various journals. From IR point of view, this task lies somewhere between classical text retrieval where the response corresponds to documents (or references to these documents) and question/answering in which the output consists of very short passages extracted from documents. The definition of a "passage" is not of course very precise.

When defining what might constitute a passage, the IR literature refers to a variety of passage types, mainly those based on delimiters such as text, window or semantic markers. For the first type, passage boundaries are

usually defined by markers supplied by the author [2] (e.g., punctuation marks, empty lines, indentations, etc.), as defined by paragraphs, sections or even sentences. Identifying these information units could be greatly facilitated by the use of XML markup languages. For the window-based type, passages have a fixed length in terms of the number of words or bytes. As a dynamic alternative to this approach, Kaszkiel & Zobel [3] suggest arbitrary passages starting at any given position within a document. For the latter, passages may also be defined according to a text's subject or semantic content. The main idea is to divide documents into logical units, with each unit being related to a single subtopic. For more detailed information on passages, a survey can be found in [4].

In order to retrieve pertinent passages two principal strategies may be used. In the first, known as *dynamic passage retrieval*, passages are defined and retrieved at search time. In this case, specific scores are assigned during a query to portions of text and thus the exact passage presented to the user depends on the query parameters submitted. As a second strategy, known as *static passage retrieval*, the elementary parameters assigned to index and search passages are predefined and identified. This is the one we adopted this year.

In elaborating the protocol for the Genomic task, the organizers limited the passage definitions. Given that the scientific articles available are in HTML format, the organizers required that a passage could be one or more sentences, as long as they did not cross the HTML paragraph tag (<P>). This type of structural constraint means passages extending beyond the paragraph tag (<P>) would not be retrieved.

Finally, in an effort to develop an effective search strategy for the biomedical publication domain, we decided to take this underlying domain into account. In this domain both the type of information and the underlying terminology evolve very rapidly. It is also well known that several names and symbols exist to denote the same protein or gene. Thus for this evaluation campaign, we decided to focus only on search terms and their possible orthographic variations in order to provide a partial solution to this problem. Once a request is submitted to our system, it will automatically add all the orthographic variants (up to 10) of the search terms (or sequence of two

search terms) found in the corpus. External resources such as gene ontologies or biomedical databases were not used this year.

The rest of this paper is organized as follows. Section 2 depicts the main characteristics of our test-collection and how the passages are derived from an article according to our passage definition. Section 3 describes the indexing approach adopted to assign descriptors to passages and Section 4 briefly presents the two probabilistic models used to score these passage representations. Section 5 describes the approach we use to handle orthographic variations. Section 6 evaluates the two IR models by applying different conditions. Finally, the main findings of this paper are presented in Section 7.

2. TEST-COLLECTION

The document collection used this year contains approximately 12 GB of uncompressed data, made up of 162,259 full-text publications extracted from 49 biomedical journals. In accordance with the general approach adopted to retrieve passages, we first processed each article to generate its corresponding passages. As passage delimiters, we made use of the following HTML tags: H1, H2, H3, H4, H5, H6, P, BR, HR, TABLE, TD, TH, TR, OL, and UL.

```

<PASSAGE>
<FN> /raid/Genomics/peds/12118078.html
<ID> 12118078.23
<SO> 28541
<L> 978
<TGN> p
<R> false
<TITLE> Alterations in the Mouse and Human
Proteome Caused by Huntington's disease
<TX> In addition to the cytoplasmic brain
fraction that was used in the above experiments,
proteins solubilized by urea and detergent
treatment, yielding an extract enriched in
membrane proteins, as well as DNA-binding
proteins released by DNase, were screened to
expand the range of protein classes studied. In
both fractions no additional proteins were
consistently different between R6/2 and control
mice (data not shown). AAT was present at low
amounts in the membrane fraction and
undetectable in the fraction of proteins
released by DNase in control mice, arguing for a
mainly cytoplasmic localization of the protein
(data not shown). ABC was found in all three
fractions. A consistently lower expression of
ABC and AAT expression below the detection limit
were detected in R6/2 samples in all three
fractions (data not shown).
</PASSAGE>

```

Figure 1. Example of generated passage

Figure 1 shows an example of a passage that might be generated. All our passages are structured according to the following set of fields:

- FN (article filename path),
- ID (passage identifier),

- SO (start offset),
- L (passage length in bytes),
- TGN (tag name from which the passage was extracted),
- R (indicates whether or not the passage is identified as a reference),
- TITLE (title of article),
- TX (passage contents).

We first established a basic rule to determine whether or not a passage would be considered a reference. As such, all passages appearing after a single line containing the word “References”, “Bibliography” or “Literature” were marked as references. After filtering all passages containing fewer than 10 words, the resulting collection contained exactly 10,700,925 passages from which 1,275,132 (11.9%) were marked as references. Table 1 lists some of the statistics on our passages, and from this we can observe that the median is clearly smaller than the mean.

| | Collection set | | Relevant set |
|--------------|----------------|----------|--------------|
| # Passages | 10,700,925 | | 3,451 |
| # References | 1,275,132 | | NA |
| Length | in words | in bytes | |
| Mean | 43.6 | 974.9 | 399.8 |
| Median | 30 | 575 | 229 |
| Std. dev. | 52.9 | 1,729.0 | 489.5 |
| Minimum | 0 | 0 | 27 (#172) |
| Maximum | 4,863 | 237,885 | 6,928 (#169) |

Table 1. Passage collection statistics

Within this collection there are 28 topics corresponding to real information needs commonly expressed by biologists. The sample text showed in Figure 2 was selected from last year’s topic set and reformulated as a simple question, delimited by the <QUESTION> tag. This topic set is subdivided into four different main scenarios (or Generic Topic Types). Regardless of the topic, the IR system will return the same type of answer, namely a ranked list of “passages”.

```

<TOPIC>
<ID> 125
<NEW-ID> 171
<GENE> Nurr-77
<PROCESS> preventing auto-immunity by deleting
reactive T-cells before they migrate to the
spleen or the lymph nodes
<QUESTION> How does Nurr-77 delete T cells before
they migrate to the spleen or lymph nodes and
how does this impact autoimmunity?
</TOPIC>

```

Figure 2. Example of a topic

Based on relevance assessments made on this test collection, the mean number of relevant passages per

topic is 132.73 (median: 35; standard deviation: 188.17). Topic #187 (“How do mutations in familial hemiplegic migraine ...?”) returned only three pertinent passages while Topic #172 (“How does p53 affect apoptosis?”) produced the greatest number of relevant passages (593). Topics #173 (“How do alpha7 nicotinic receptor subunits affect ethanol metabolism?”) and #180 (“How do Ret-GDNF interactions affect liver development?”) did not reveal any relevant passages and were thus discarded from the evaluation.

3. INDEXING APPROACHES

As a natural approach to indexing and searching into our defined passages, we chose words as the indexing units. Based on this scheme, our lexical analyzer applies the followings steps to process the input. First, the text is tokenized (using spaces or punctuation marks), simple acronyms are normalized (e.g., P.S.A. is converted into PSA) and hyphenated terms are also broken up into their components. For example, a word such as “COUP-TF1” generates three different forms, namely “COUP”, “TF1” and the original form “COUP-TF1”.

Second, uppercase letters are transformed into their lowercase forms. Third, stopwords are filtered out using the SMART list (571 entries). Fourth, the *S-stemmer* algorithm [5] based on three rules removes the final ‘-s’ (the most common plural suffix for the English language).

In our experiments last year [6], we showed that among the four evaluated stemmers (Lovins, *S-stemmer*, Porter and SMART) the *S-stemmer* achieved the best retrieval effectiveness.

As an alternative and in order to reduce the negative impact caused by spelling errors or orthographic variation, we adopted the 5-gram as a second indexing approach. This method does not require any prior linguistic knowledge and is also more robust in handling typographical errors, both in the submitted query and in the documents retrieved. Within this context, we adopt an overlapping 5-gram approach. For instance, for the term “alzheimer” the system automatically generates the following 5-gram variants: “alzhe”, “lzhei”, “zheim”, “heime” and “eimer”.

Both topics and documents are processed in the same way. However, for topics only and based on [7], we apply an extended stopword list comprising seven additional terms (namely gene, impact, method, role, biological, disease and process). These words occur very frequently in the biomedical domain and thus are not helpful in discriminating between relevant and non-relevant documents. Finally, in order to hopefully improve the retrieval effectiveness, we automatically include the document title in all tile passages generated from a given scientific article (listed under the <TITLE>

tag in Figure 1). Of course as shown in Section 6, it is also possible to ignore this indexing feature.

4. GENERATION OF ORTHOGRAPHIC VARIANTS

As is known, in biomedical literature several orthographic variants [7] can be found to represent a given name and these are generally introduced for a variety of reasons:

- 1) Typographic errors and misspellings (e.g. “retrieval” and “retrieval”) or cognitive (e.g., “ecstasy”, “extasy”, or “ecstasy”; “occurrence” or “occurrence”);
- 2) Alternative punctuation and tokenization, mainly due to the lack of a naming convention (e.g. “Nur77”, “Nurr-77” or “Nurr 77”);
- 3) Regional language variations, such as British and American English (e.g. “colour” or “color”, “grey” or “gray”, etc.)
- 4) Transliteration of foreign names (e.g., “Crohn” and “Krohn” or “Creutzfeld-Jakob” and “Creutzfeldt-Jacob”);
- 5) Morphological variations (inflections or derivations) which could be resolved by using a stemmer.

During previous TREC campaigns, many methods were proposed to resolve the problem of orthographic variations, as for example [8]. The algorithms proposed were usually rule-based and were essentially concerned with secondary causes described above (e.g., see [9]).

In order to automatically find a ranked list of alternative spellings for each search word, we modified the Lucene [10] Spell Checker¹. In its initial stage this tool required a lexicon containing the correct spelling, so in our case we used the words extracted from the TREC 2005 corpus, a large subset of the MEDLINE collection. We then introduced a single term or a short sequence of words, limited in the current case to two terms. In response, the spell checker returned a ranked list of the top 100 hits extracted from the given lexicon. In our case we re-ranked this list according to the minimal *edit-distance* measure and its length, for each candidate that was a variant of the original (misspelled) term submitted as follows:

$$\text{Score} = 1 - [\text{edit-distance} / \text{length}(\text{term})]$$

When the two similar candidates were deemed to be equal (which occurred relatively frequently), they were ordered according to popularity (or *df*, document frequency), ranging from most to less frequent.

¹ <http://wiki.apache.org/jakarta-lucene/SpellChecker>

For each topic available in this TREC campaign, we submitted each search word or group of two successive words to the spellchecker engine. As shown in Figure 3, the top ten suggested spelling candidates, which were then re-sequenced by the *edit* and *df* measure, and automatically added to topic following the <S> tag (followed by the alternative number).

```

<TOPIC>
<ID> 125
<NEW-ID> 171
<GENE> Nurr-77
<PROCESS> preventing auto-immunity by deleting
reactive T-cells before they migrate to the
spleen or the lymph nodes
<QUESTION> How does Nurr-77 delete T cells before
they migrate to the spleen or lymph nodes and
how does this impact autoimmunity?
<s1 input="nurr 77" df="1">nurr-77
<s2 input="nurr 77" df="28">nurr77
<s1 input="nurr-77" df="41">nur-77
<s2 input="nurr-77" df="28">nurr77
...
<s1 input="auto immunity" df="32">auto-immunity
<s2 input="auto immunity" df="6527">autoimmunity
...
<s1 input="lymph node" df="202">lymph-node
<s2 input="lymph node" df="4">lymphnode
<s3 input="lymph node" df="38">lymphnode
</TOPIC>

```

Figure 3. Example of topic with orthographic variants

In Figure 3, the *input* attribute describes what was submitted to the spellchecker. The *df* attribute indicates the number of passages indexed under the suggested variant. For example, the first orthographic variant proposed by the spell checker for the sequence “nurr 77” is “nurr-77” and its popularity is 1 (the proposed term “nurr-77” occurs only in one passage). A more complete example is given in the Appendix, Figure 4. In our experiments we did not use the popularity information for weighting additional search term and we only considered the top ten orthographic alternatives.

5. RETRIEVAL MODELS

In our evaluations, we considered two probabilistic retrieval models. As a first approach, we used the Okapi (BM25) model [11] in which the score of the document D_i for the current query Q was evaluated using the following formula:

$$Score(D_i, Q) = \sum_{t_j \in q} qtf \cdot \log \left(\frac{n - df_j}{df_j} \right) \cdot \frac{(k_1 + 1) \cdot tf_{ij}}{K + tf_{ij}}, \quad (1)$$

where $K = k_1 \cdot [(1 - b) + b \cdot (l_i / avdl)]$

As a second approach, we used the I(n)B2 model derived from the *Divergence from Randomness* (DFR) paradigm [12]. In this case, the document score is evaluated as:

$$Score(D_i, Q) = \sum_{t_j \in q} qtf \cdot w_{ij} \quad (2)$$

where the weight w_{ij} of term t_j in document D_i is based on combining two information measures as follows:

$$w_{ij} = Inf_{ij}^1 \cdot Inf_{ij}^2$$

$$Inf_{ij}^1 = -\log_2 prob_{ij}^1 = tfn \cdot \log_2 \left[\frac{(n+1)/(df_j + 0.5)}{(tc_j + 1)/(df_j \cdot (tfn_{ij} + 1))} \right] \quad (3)$$

$$Inf_{ij}^2 = 1 - prob_{ij}^2 = (tc_j + 1)/(df_j \cdot (tfn_{ij} + 1)) \quad \text{and}$$

$$tfn_{ij} = tf_{ij} \cdot \log_2(1 + c \cdot (avdl/l_i))$$

in which $prob_{ij}^1$ is the pure chance probability of having tf_{ij} occurrences of the term t_j in a document. On the other hand, $prob_{ij}^2$ is the probability of encountering a new occurrence of term t_j in the document, given that we already found tf_{ij} occurrences of this term.

Within these two retrieval models, *qtf* denotes the frequency of term t_j in query Q , *df_j* indicates the number of documents indexed with the term t_j , *n* the number of documents in the corpus, *tc_j* the number of occurrences of term t_j in the collection, *l_i* is the length (number of indexing terms) of document D_i , *avdl* is the average document length, and *k₁*, *b* and *c* are constants. In our experiments, we set the values of *c*, *k₁*, *b* and *avdl* to be 5.0, 1.2, 0.55 and 146 respectively for the main field content (<P>) and 1.5, 1.2, 0.55 and 61 for the title (<TITLE>) field. These values were chosen according to our experiments, based on the Genomic TREC-2005.

6. EVALUATION

To evaluate our various search strategies, we used the tool provided, that is TRECGEN2006_SCORE. Based on the retrieval of 1,000 passages per query, this program computed the performance at three different levels: 1) mean average precision (MAP); 2) mean average passage precision (MAPP); and 3) mean average aspect precision (MAAP). To statistically determine whether or not a given search strategy would be better than another, we applied the *t*-test [13]. In our statistical tests, the null hypothesis H_0 stated that both retrieval schemes resulted in similar retrieval performance. Therefore in the experiments presented in this paper, statistically significant differences were detected by a two-sided *t*-test (significance level $\alpha = 5\%$).

6.1 Indexing Strategies

Table 2 depicts the various evaluations achieved by the word-based indexing approach (columns 2 to 4), by the 5-gram indexing scheme (columns 5 to 7), as well as the relative performance difference between the word and the 5-gram indexing strategies. This table also lists the best performance under a given condition (depicted in bold), which will be used as the baseline for statistical testing.

| \ Index | word | | | 5-gram | | | % difference (5-gram vs. word) | | |
|----------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|--------------------------------|---------|----------------|
| Measure | MAP | MAPP | MAAP | MAP | MAPP | MAAP | MAP | MAPP | MAAP |
| BASE | 0.3734 | 0.0418 | 0.2573 | 0.3351 | 0.0343 | 0.1962 | -10.26% | -17.94% | -23.75% |
| +SPELL | 0.3642 (-2.46%) | 0.0404 (-3.35%) | 0.2538 (-1.36%) | 0.3106 (-7.31%) | 0.0337 (-1.75%) | 0.1886 (-3.87%) | -14.72% | -16.58% | -25.69% |
| +TITLE | <u>0.2943</u> (-21.18%) | <u>0.0328</u> (-21.51%) | <u>0.2027</u> (-21.22%) | <u>0.2489</u> (-25.72%) | <u>0.0230</u> (-32.94%) | 0.1294 (-34.05%) | -15.42% | 29.88% | <u>-36.16%</u> |
| +TITLE & SPELL | <u>0.3017</u> (-19.20%) | <u>0.0322</u> (-22.97%) | <u>0.2027</u> (-21.22%) | <u>0.2257</u> (-32.65%) | <u>0.0225</u> (-34.40%) | <u>0.1175</u> (-40.11%) | <u>-25.19%</u> | -30.12% | <u>-42.03%</u> |

Table 2. Performance of the I(n)B2 model under various schemes (26 queries)

| \ Index | word | | | 5-gram | | | % difference (5-gram vs. word) | | |
|----------------|----------------------------|-----------------------------|---------------------|----------------------------|---------------------|----------------------------|--------------------------------|---------|---------|
| Measure | MAP | MAPP | MAAP | MAP | MAPP | MAAP | MAP | MAPP | MAAP |
| BASE | 0.2426 | 0.0270 | 0.1248 | 0.2624 | 0.0233 | 0.1262 | +8.18% | -13.70% | +1.12% |
| + SPELL | 0.2422 (-0.16%) | 0.0265 (-1.85%) | 0.1214 (-2.72%) | 0.2310 (-11.97%) | 0.0188 (-19.31%) | 0.1032 (-18.23%) | -4.62% | -29.06% | -14.99% |
| + TITLE | <u>0.1827</u> (-24.69%) | <u>0.0197*</u> (-27.04%) | 0.1075 (-13.86%) | <u>0.1740</u> (-33.69%) | 0.0157 (-32.62%) | 0.0890 (-29.48%) | -4.76% | -20.30% | -17.21% |
| +TITLE & SPELL | <u>0.1776</u> (-26.79%) | <u>0.0192</u> (-28.89%) | 0.1081 (-13.38%) | <u>0.1519</u> (-42.11%) | 0.0141 (-39.48%) | <u>0.0700</u> (-44.53%) | -14.47% | -26.56% | -35.25% |

Table 3. Performance of the Okapi model under various schemes (26 queries)

The lines in this table represent various features that we used during the indexing and search process.

First, we wanted to verify the relative performance of our suggested orthographic variants generation. The third line contains the baseline approach while the line labeled “+SPELL” depicts the retrieval performance for the orthographic variation. In the fourth line (labeled “+TITLE”) the article’s title is always included for each passage (as shown in Figure 1). Finally, the performance achieved using both the article’s title and the orthographic variants are depicted (label “+TITLE & SPELL”).

Table 3 shows the same information using the Okapi probabilistic model. From these two tables, the following conclusions can be drawn. The overall best retrieval performance is always obtained by the simplest system, without considering the article’s title or the orthographic variations. The performance difference is relatively small (around -2.2%) when we do or do not include the orthographic variants of query terms. On the other hand, taking the article’s title into account for clearly hindered the retrieval performance (the differences are always statistically significant and therefore underlined in Tables 2 and 3).

Moreover, when comparing word-based indexing strategy with the 5-gram, the word-based approach usually has the best retrieval performance. The only exception to this rule was the performance achieved with the simplest model, using the Okapi approach (word: 0.2426, 5-gram: 0.2624).

6.2 Official Runs

Table 4 lists the evaluation results for our three official runs, together with their various components. Our runs are based on the two probabilistic models including some of the search features described previously. Thus in this table we first considered the I(n)B2 model with and without orthographic variants (lines 1 and 2), and the same model based on the 5-gram approach (line 3). For the Okapi model (word-based, line 4), we also considered the orthographic variants (line 5).

The run labeled “UniNE1” combines both the I(n)B2 model (word-based and 5-gram) and the Okapi approach (word-based). As a data fusion approach, we used the z-score method [14]. Within this scheme, we normalized document scores (or passage score in the current case) for each D_k provided by the i th result list, as computed by:

$$Z\text{-score } RSV_k = [((RSV_k - \text{Mean}^i) / \text{Stdev}^i) + \delta^i],$$

$$\delta^i = ((\text{Mean}^i - \text{Min}^i) / \text{Stdev}^i) \quad (6)$$

within which Mean^i denotes the average of the RSV_k , and Stdev^i the standard deviation.

The second official run “UniNE2” was also based on the combination of different search strategies. In this case, we generated the I(n)B2 with the article title included in each passage (line 7), or with both the article title and orthographic variants (line 8). The result lists are combined using a simpler normalization procedure (for each result list, each passage score was divided by the maximum score).

| IR Models | MAP | MAPP | MAAP |
|--|---------------|---------------|---------------|
| 1. I(n)B2 (word) | 0.3734 | 0.0418 | 0.2573 |
| 2. I(n)B2 (word) + orthographic variants | 0.3642 | 0.0404 | 0.2538 |
| 3. I(n)B2 (5-GRAMS) | 0.3351 | 0.0343 | 0.1962 |
| 4. Okapi (word) | 0.2426 | 0.0270 | 0.1248 |
| 5. Okapi (word) + orthographic variants | 0.2422 | 0.0265 | 0.1214 |
| 6. Data fusion (2, 3 & 5), Z-SCORE (UNINE1) | 0.3539 | 0.0390 | 0.2070 |
| 7. I(n)B2 (word) with T field | 0.2943 | 0.0328 | 0.2027 |
| 8. I(n)B2 (word) with T field + variants | 0.3017 | 0.0322 | 0.2027 |
| 8. Data fusion (8, 3 & 5), NORM rsv (UNINE2) | 0.3460 | 0.0384 | 0.2018 |
| 9. Data fusion (1 & 2), Z-SCORE (UNINE3) | 0.3725 | 0.0407 | 0.2259 |

Table 4. Results of our official runs and their components

Finally, the third run “UniNE3” was a simple combination of the I(n)B2 with and without the orthographic variations. An overall view of Table 4 indicates that “simpler approaches are more effective than complex ones”, just as we concluded last year [6].

7. CONCLUSION

During the TREC 2006 Genomic evaluation campaign we evaluated various indexing and search strategies. The empirical evidence collected shows that a word-based approach performs better than a 5-gram indexing scheme (relative difference around 15%). This comes as a surprise, given that the 5-gram approach is usually more robust than the word-based scheme. The inclusion of orthographic variants for search words (or two-word query sequences) does not improve retrieval effectiveness, at least as implemented in our system. When comparing the I(n)B2 model derived from the *Divergence from Randomness* paradigm with that of the Okapi approach in which various indexing strategies (word or 5-gram) are considered, the resultant MAP is better (by about 50%) and in favor of the I(n)B2 model.

The generation of passages delimited by HTML tags was not a success. The performance achieved was in fact rather poor, suggesting that there were too many sentences within our text passages. As another extreme alternative, we might generate one passage per sentence, yet for the moment results for this passage definition remains unknown.

From an IR point of view, the automatic inclusion of the article title in each passage is not effective. In all cases studied during this evaluation campaign, the inclusion of this logical element actually hinders retrieval performance.

ACKNOWLEDGMENTS

This research was supported in part by the Swiss NSF under Grant #200020-103420.

8. REFERENCES

- [1] Hersh, W.R., Cohen, A.M., Yang, J., Bhuptiraju, R.T., Roberts, P., & Hearst, M. TREC 2005 genomics track overview. In *Proceedings of TREC-2005*. Gaithersburg, MA, 2006.
- [2] Salton, G., Allan, J., & Buckley, C. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th annual international ACM-SIGIR conference on research and development in information retrieval*, Pittsburgh, PA, 1993, 49-58.
- [3] Kaszkiel, M., & Zobel, J. Passage retrieval revisited. In *Proceedings of the 20th annual international ACM-SIGIR conference on research and development in information retrieval*, Philadelphia, PA, 1997, 178-185.
- [4] Kaszkiel, M., & Zobel, J. Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology*, 52(4), 2001, 344-364.
- [5] Harman, D. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1), 1991, 7-15.
- [6] Abdou, S., Ruch, P., & Savoy, J. Evaluation of stemming, query expansion and manual indexing approaches for the Genomic task. In *Proceedings TREC-2005*, Gaithersburg, MA, 2006, 863-871.
- [7] Yu, H., & Agichtein, E. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19(1), 2003, i340-i349.
- [8] Huang, X., Zhong, M., & Si, L. York University at TREC 2005: Genomics Track. In *Proceedings of TREC-2005*. Gaithersburg, MA, 2006.
- [9] Cohen, A.M. Unsupervised gene/protein named entity normalization using automatically extracted

dictionaries. In *Proceeding ACL-ISMB*, Detroit (MI), 2005, 17-24.

- [10] Gospodnetic, O., & Hatcher, E. *Lucene in Action*. Manning Publications, 2004
- [11] Robertson, S.E., Walker, S., & Beaulieu, M. Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1), 2000, 95-108.
- [12] Amati, G., & van Rijsbergen, C.J. Probabilistic models of information retrieval based on measuring

the divergence from randomness. *ACM-Transactions on Information Systems*, 20(4), 2002, 357-389.

- [13] Conover, W.J. *Practical Nonparametric Statistics*. 3rd edn. John Wiley & Sons, 1999.
- [14] Savoy, J. Data fusion for effective European monolingual information retrieval. In C. Peters, P.D. Clough, J. Gonzalo, G.J.F. Jones, M. Kluck & B. Magnini (Eds.), *Multilingual Information Access for Text, Speech and Images*. Lecture Notes in Computer Science #3491. Springer-Verlag, Berlin, 2005, 233-244.

```
<TOPIC>
<ID> 125
<NEW-ID> 171
<GENE> Nurr-77
<PROCESS> preventing auto-immunity by deleting reactive T-cells before they migrate to the
spleen or the lymph nodes
<QUESTION> How does Nurr-77 delete T cells before they migrate to the spleen or lymph nodes and
how does this impact autoimmunity?
<s1 input="nurr 77" score="0.86" df="1"> nurr-77
<s2 input="nurr 77" score="0.83" df="28"> nurr77
<s1 input="nurr-77" score="0.83" df="41"> nur-77
<s2 input="nurr-77" score="0.83" df="28"> nurr77
<s1 input="preventing" score="0.80" df="31802"> prevention
<s1 input="auto immunity" score="0.92" df="32"> auto-immunity
<s2 input="auto immunity" score="0.92" df="6527"> autoimmunity
<s3 input="auto immunity" score="0.85" df="28"> autoimmunity
<s4 input="auto immunity" score="0.85" df="1"> autoimmuinity
<s6 input="auto immunity" score="0.83" df="1"> autoimmunita
<s7 input="auto immunity" score="0.82" df="1"> utoimmunity
<s8 input="auto immunity" score="0.82" df="1"> autommunity
<s1 input="auto-immunity" score="0.92" df="6527"> autoimmunity
<s1 input="reactive t" score="0.80" df="14"> reactively
<s2 input="reactive t" score="0.80" df="1"> reactiveyy
<s1 input="cell t-cell" score="0.91" df="15"> cell-t-cell
<s2 input="cell t-cell" score="0.82" df="4366"> cell-to-cell
<s3 input="cell t-cell" score="0.82" df="3"> cell-b-cell
<s4 input="cell t-cell" score="0.82" df="3"> cells-t-cell
<s5 input="cell t-cell" score="0.82" df="3"> cell-to-cell
<s6 input="cell t-cell" score="0.82" df="1"> cellto-cell
<s1 input="lymph node" score="0.90" df="202"> lymph-node
<s2 input="lymph node" score="0.90" df="4"> lymphonode
<s3 input="lymph node" score="0.89" df="38"> lymphnode
<s1 input="nurr 77" score="0.86" df="1"> nurr-77
<s2 input="nurr 77" score="0.83" df="28"> nurr77
<s1 input="delete" score="0.83" df="32760"> deleted
<s1 input="cell migrate" score="0.83" df="1"> cell-migrated
<s1 input="lymph node" score="0.90" df="202"> lymph-node
<s2 input="lymph node" score="0.90" df="4"> lymphonode
<s3 input="lymph node" score="0.89" df="38"> lymphnode
</TOPIC>
```

Figure 4: Full example of topic with its orthographic variants