Enhancing the TDT Tracking Evaluation

Amit Bagga

GE Corporate Research and Development
1 Research Circle
Niskayuna, NY 12309. USA
bagga@crd.ge.com

Abstract

Topic Detection and Tracking (TDT) is a DARPA-sponsored initiative concerned with finding groups of stories on the same topic (tdt, 1998). The goal is to build systems that can segment, detect, and track incoming news stories (possibly from multiple continuous feeds) with respect to pre-defined topics. While the detection task detects the first story on a particular topic, the tracking task determines, for each story, which topic it is relevant to. This paper will discuss the algorithm currently used for evaluating systems for the tracking task, present some of its limitation, and propose a new algorithm that enhances the current evaluation.

1. Introduction

Topic Detection and Tracking (TDT) is a DARPA-sponsored initiative concerned with finding groups of stories on the same topic (tdt, 1998). The goal is to build systems that can segment, detect, and track incoming news stories (possibly from multiple continuous feeds) with respect to pre-defined topics. While the detection task detects the first story on a particular topic, the tracking task determines, for each story, which topic it is relevant to. In more precise terms, the tracking task is specified as follows: given N_t training stories on a topic, the system must find all subsequent stories on the same topic in all tracked news sources. These sources include radio and television news broadcasts, as well as newswire feeds. The initial set of training stories (usually 1, 2, or 4) is the only information about the topic available to the tracking system. This paper will discuss the algorithm currently used for evaluating systems for the tracking task, present some of its limitation, and propose a new algorithm that enhances the current evaluation.

2. Description of the Current Tracking Evaluation Algorithm

The current algorithm measures the performance of systems in terms of miss and false alarm (FA) rates. A "miss" occurs when, for a particular topic, the system identifies a story to be non-relevant when it actually is relevant. A "false alarm", on the other hand, occurs when, for a particular topic, the system identifies a story as relevant when it actually is not. The computation of the miss and false alarm rates can be done in two ways:

- By computing the miss and false alarm rates without any regard to the topic – the unweighted or Pooled method.
- By computing the miss and false alarm rates such that all topics have equal weight – the Weighted method.

While the pooled method minimizes the variance caused by individual story decisions, the weighted method has the advantage of minimizing the variance of the estimates caused by topic differences. Because of the small number of topics in TDT-2, and because of topic inhomogeneity, the weighted method was actually used for that evaluation. We describe, in detail, each of these methods.

2.1. The Unweighted or Pooled Method

The unweighted tracking evaluation algorithm measures the miss and the false alarm rates as shown in the equations in Figure 1. In these equations, the summation is over all topics, where $Stories_t$, for each topic t, is the number of non-training stories to be tracked, and where $\delta_{sys}(t,s)$ is as shown in Figure 2. Similar to $\delta_{sys}(t,s)$, $\delta_{ref}(t,s)$ is 1 if the answer key (or, the truth) deemed that topic *t* was discussed in story *s*, and 0 otherwise. A more detailed explanation of the equations in Figure 1 follows.

As mentioned earlier, P_{Miss} estimates the percentage of times the system identifies a relevant story as being non-relevant. Therefore, the denominator in the P_{Miss} equation counts, for each topic, the number of stories that are actually relevant to that topic since $\delta_{ref}(t,s)$ is 1 *iff* topic t is mentioned in story s. Computing the sum over all topics yields the total number of stories, in the collection, that are relevant to at least one topic. It should be noted that the TDT evaluation enforces that a story is relevant to only one topic.

In the numerator of the equation for P_{Miss}, the $(1 - \delta_{sys}(t,s)).\delta_{ref}(t,s)$ product is 1 *iff* $\delta_{sys}(t,s)$ is 0 *and* $\delta_{ref}(t,s)$ is 1. In other words, the product is 1 *iff* the system determines that t is not mentioned in s when the truth determines that it is. Therefore, computing the sum of this product over all stories and over all topics yields the total number of occurrences of a missed story.

Similarly, the denominator in the P_{FA} equation computes, for each topic, the number of stories that are not relevant to that topic. However, since a story is relevant to only one topic, when the sum is computed over all topics, each story is counted $Topics - 1$ (where *Topics* is the total number of topics) times as it is not relevant to each of these $Topics - 1$ topics. Therefore, this sum is significantly greater than the total number of stories, $Stories_t$, in the collection since, for each topic, a large number of stories are not relevant to it.

$$P_{\text{Miss}} = \sum_{t \in \textit{Topics}} \left\{ \sum_{s \in \textit{Stories}_t} \left\{ (1 - \delta_{\textit{sys}}(t,s)).\delta_{\textit{ref}}(t,s) \right\} \right\} \Big/ \sum_{t \in \textit{Topics}} \left\{ \sum_{s \in \textit{Stories}_t} \delta_{\textit{ref}}(t,s) \right\}$$

$$P_{\text{FA}} = \sum_{t \in \textit{Topics}} \left\{ \sum_{s \in \textit{Stories}_t} \left\{ \delta_{\textit{sys}}(t,s).(1 - \delta_{\textit{ref}}(t,s)) \right\} \right\} \Big/ \sum_{t \in \textit{Topics}} \left\{ \sum_{s \in \textit{Stories}_t} (1 - \delta_{\textit{ref}}(t,s)) \right\}$$

Figure 1: Equations for the Current Evaluation Algorithm

$$\delta_{\textit{sys}}(t,s) = \begin{cases} 1 & \text{if the } \textit{system} \text{ deemed that topic } t \text{ was discussed in story } s \\ 0 & \text{otherwise} \end{cases}$$

Figure 2: Definition of $\delta_{\textit{sys}}(t,s)$

The $\delta_{\textit{sys}}(t,s).(1 - \delta_{\textit{ref}}(t,s))$ product in the numerator of the P_{FA} equation is 1 *iff* $\delta_{\textit{sys}}(t,s)$ is 1, and $\delta_{\textit{ref}}(t,s)$ is 0. In other words, the product is 1 *iff* the system determines that t is mentioned in s when the truth determines that it actually is not. Therefore, computing the sum over all stories and over all topics yields the total number of occurrences of a false alarm.

The Weighted method is discussed later in the paper in Section 5.

3. Limitation of the Current Algorithm

While the evaluation algorithm has been successful in measuring the performance of systems, it has three main limitations.

1. It is possible for a system to get low miss and false alarm rates in cases where the system actually makes a lot of errors. Consider the following example. Suppose we are tracking only one topic, and the number of stories that are going to be tracked is 1000 ($S_1, S_2, \ldots, S_{1000}$). Also suppose that the truth identifies that $S_1, S_2, \ldots, S_{10}$ are the stories relevant to the topic while the others are not relevant. Now if a system responds by tracking $S_1, \ldots, S_9, S_{11}, S_{12}, \ldots, S_{101}$ as relevant to the topic while identifying the rest to be non-relevant, then

   $$P_{\text{Miss}} = \frac{1}{10} = 0.1$$

   since it only misses one story, S_{10}. In addition,

   $$P_{\text{FA}} = \frac{91}{990} = .092$$

   since the system identifies 91 ($S_{11}, \ldots, S_{101}$) of the 990 non-relevant stories as actually being relevant.

   But, the precision in this case is $9/100 = 0.09$ i.e. the system correctly tracked only 9 out of the 100 articles. The poor performance of the system in this case is not accurately reflected by the low miss and the false alarm rates which actually indicate good performance.

2. The algorithm penalizes all errors equally. Consider the following example. Suppose we have four topics $t_1, t_2, \ldots, t_4$, and that, for each topic, $\textit{Stories}_t$, the total number of stories to be tracked for that topic, is 500. Also suppose that the truth deems that $S_1, \ldots, S_{200}$ are relevant to topic t_1, $S_{201}, \ldots, S_{400}$ are relevant to t_2, $S_{451}, \ldots, S_{450}$ are relevant to t_3, and $S_{451}, \ldots, S_{500}$ are relevant to t_4. Now if a response, R_1, has a system identifying that $S_1, \ldots, S_{200}$ are relevant to t_1, $S_{201}, \ldots, S_{450}$ are relevant to t_2, and $S_{451}, \ldots, S_{500}$ are relevant to t_4, then,

   $$P_{\text{Miss}} = \frac{0 + 0 + 50 + 0}{200 + 200 + 50 + 50} = \frac{50}{500} = 0.1$$

   since the only stories missed are the 50 ($S_{401}, \ldots, S_{450}$) relevant to t_3. In addition,

   $$P_{\text{FA}} = \frac{0 + 50 + 0 + 0}{300 + 300 + 450 + 450} = \frac{50}{1500} = 0.033$$

   since the system identifies, for topic t_2, 50 stories ($S_{401}, \ldots, S_{450}$) as being relevant when they actually are not.

   But, if a response, R_2, has the system identifying that $S_1, S_2, \ldots, S_{200}$ are relevant to t_1, $S_{201}, \ldots, S_{400}$ are relevant to t_2, and $S_{401}, \ldots, S_{500}$ are relevant to t_4, then,

   $$P_{\text{Miss}} = \frac{0 + 0 + 50 + 0}{200 + 200 + 50 + 50} = \frac{50}{500} = 0.1 \text{ and,}$$

   $$P_{\text{FA}} = \frac{0 + 0 + 0 + 50}{300 + 300 + 450 + 450} = \frac{50}{1500} = 0.033$$

Therefore, the current algorithm assigns equal miss and false alarm scores for both the responses in the example above when in fact R_1 merged all the articles corresponding to a large topic (in the number of documents relevant), t_2, and a small topic, t_3, together while R_2 merged all documents corresponding to the two small topics, t_3, and t_4, together. We feel that R_1 should have been penalized more heavily than R_2 because R_1 implicitly creates a larger number of errors by linking the 200 document topic with the 50 document topic.

3. The False Alarm measure deflates the actual error rate because for each story it credits the system for not making a mistake multiple times. This happens because the algorithm considers all non-relevant stories when computing the false-alarm rate. Since a story is relevant to only one topic (i.e. it is not relevant to the remaining topics), the system is credited for not making a mistake for each of the $t-1$ topics (if t is the total number of topics). The deflation actually occurs because of the denominator of the equation that calculates P_{FA}. As mentioned earlier, the value of the denominator is usually significantly greater than the total number of stories in the collection since, for each topic, a large number of stories are not relevant to it. Therefore, computing this sum over all topics yields a rather large number compared to the number of stories. For instance, in the example described above, the denominator for P_{FA} is 1500, or three times the total number of stories.

4. The B-CUBED Algorithm

The B-CUBED algorithm is based upon an algorithm developed to score coreference chains(Bagga and Baldwin, 1998). It was designed specifically to identify and penalize both explicit as well as implicit errors. The algorithm measures performance in terms of precision and recall. These are defined as shown in Figure 3. In the equations shown in the figure, *ST* is the set of stories being tracked, *Topics* is the set of topics in the response, and δ_{ref}, and δ_{sys} are as defined earlier for the current algorithm.

Alternatively, for each story, s, Precision_s, and Recall_s can be explained, in words, as in Figure 4. The final precision and recall numbers are computed by taking an average of P_s, and R_s, for all stories s in *ST*.

The B-CUBED algorithm computes precision and recall with respect to the stories tracked. As shown in Figure 4, the algorithm computes the numerator of the precision equation, for each story s, by analyzing the topic that s is actually relevant to (as deemed by the truth), and computing the number of stories in common with the topic in the response containing s. The denominator of the equation is computed by counting the total number of stories relevant to the topic in the response that s is relevant to. While the numerator of the recall equation is the same as that of the precision equation, the denominator is computed by counting the number of stories relevant to the topic in the truth that s is relevant to.

The equations in Figure 4 are almost equivalent to the formulae in Figure 3 the only difference being that the formulae in the latter figure compute final precision and recall. In the formula for precision, $\delta_{sys}(t,s)$ has value 1 only when t is instantiated to the topic that the system deems s relevant to. For all other topics it has value 0. Suppose the topic that the system determines that s is relevant to is topic x, then the denominator of the fraction computes the total number of stories that are relevant to topic x, as determined by the response. The numerator of the fraction, on the other hand, is the sum of the products of three expressions: $\delta_{ref}(t_1,s)$, $\delta_{ref}(t_1,s_1)$, and $\delta_{sys}(t,s_1)$. Therefore, the numerator counts the number of times each of these three expressions simultaneously have value 1. $\delta_{ref}(t_1,s)$ has a value 1 when t_1 is the topic that s actually is relevant to, say topic y (note: if the response is correct, then y is the same as x). Therefore, $\delta_{ref}(t_1,s_1)$ is 1 *iff* s_1 is one of the stories relevant to topic y in the truth. Finally, $\delta_{sys}(t,s_1)$ is 1 *iff* the story that is relevant to topic y in the truth is also relevant to topic x in the response. Thus, the numerator of the fraction counts, for each story s, the number of stories in common between the truth and the response topics containing s.

For recall, the numerator of the fraction is the same as the numerator of the fraction in the formula for precision. However, the denominator is the sum of the products of two expressions: $\delta_{ref}(t_1,s)$, and $\delta_{ref}(t_1,s_1)$. Therefore, the denominator counts the number of times both these expressions simultaneously have value 1. $\delta_{ref}(t_1,s)$ is 1 *iff* t_1 is the topic that s is actually relevant to (topic y). Moreover, $\delta_{ref}(t_1,s_1)$ is 1 *iff* s_1 is a story that is actually relevant to topic y. Therefore, the product counts the total number of stories that are relevant to the topic that s actually is relevant to.

Consider, for example, the response R_1 presented earlier while describing the second limitation of the current algorithm. The precision for this response would be calculated in the following way. $P_1 = P_2 = \ldots = P_{200} = \frac{200}{200}$ since for each of these stories, s, the system identifies 200 stories as relevant to the topic that s is actually relevant to (t_1). However, $P_{201} = \ldots = P_{400} = \frac{200}{250}$ since for each of these 200 stories, s, only 200 out of the 250 stories are actually stories in the topic that s is actually relevant to (t_2). Similarly, $P_{401} = \ldots = P_{450} = \frac{50}{250}$ since for each of these stories only 50 stories are relevant to the topic that the story itself is actually relevant to (t_3). Finally, $P_{451} = \ldots = P_{500} = \frac{50}{50}$ since for each of these stories, s, the system identifies the 50 stories in the topic that s is actually relevant to (t_4). Therefore, the final precision, the average of the precisions for the 500 stories, equals

$$\frac{1}{500}\left\{200 * \frac{200}{200} + 200 * \frac{200}{250} + 50 * \frac{50}{250} + 50 * \frac{50}{50}\right\}$$

$$= \frac{200 + 160 + 10 + 50}{500} = \frac{420}{500} = 0.84.$$

In comparison, the precision for R_2 would equal

$$\frac{1}{500}\left\{200 * \frac{200}{200} + 200 * \frac{200}{200} + 50 * \frac{50}{100} + 50 * \frac{50}{100}\right\}$$

$$P = \frac{1}{ST}\left[\sum_{s\in ST}\sum_{t\in Topics}\delta_{sys}(t,s) * \left(\frac{\left\{\sum_{s_1\in ST}\sum_{t_1\in Topics}\delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1) * \delta_{sys}(t,s_1)\right\}}{\sum_{s_1\in ST}\delta_{sys}(t,s_1)}\right)\right]$$

$$R = \frac{1}{ST}\left[\sum_{s\in ST}\sum_{t\in Topics}\delta_{sys}(t,s) * \left(\frac{\left\{\sum_{s_1\in ST}\sum_{t_1\in Topics}\delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1) * \delta_{sys}(t,s_1)\right\}}{\sum_{s_1\in ST}\sum_{t_1\in Topics}\delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1)}\right)\right]$$

Figure 3: Equations for the B-CUBED Algorithm

$$\mathrm{P_s} = \frac{\#\ of\ stories\ in\ common\ between\ topic\ that\ system\ deems\ s\ rel\ to\ and\ topic\ that\ s\ is\ \mathbf{actually}\ rel\ to}{total\ number\ of\ stories\ deemed\ by\ system\ as\ relevant\ to\ topic\ that\ it\ deems\ s\ is\ relevant\ to}$$

$$\mathrm{R_s} = \frac{\#\ of\ stories\ in\ common\ between\ topic\ that\ system\ deems\ s\ rel\ to\ and\ topic\ that\ s\ is\ \mathbf{actually}\ rel\ to}{total\ number\ of\ stories\ deemed\ by\ \mathbf{truth}\ as\ relevant\ to\ topic\ that\ s\ is\ \mathbf{actually}\ relevant\ to}$$

Figure 4: Descriptions of $\mathrm{Precision_s}$, and $\mathrm{Recall_s}$ for the B-CUBED Algorithm

$$= \frac{200+200+25+25}{500} = \frac{450}{500} = 0.9.$$

In addition, if there was a third response, R_3 which merged the two large topics, t_1 and t_2, together, then the precision for R_3 would equal

$$\frac{1}{500}\left\{200 * \frac{200}{400} + 200 * \frac{200}{400} + 50 * \frac{50}{50} + 50 * \frac{50}{50}\right\}$$

$$= \frac{100+100+50+50}{500} = \frac{300}{500} = 0.6.$$

On the other hand, the recall for all the three responses would equal 1 as the system, for each of the responses, groups all the articles belonging to the same topic together. In other words, for each of the topics, all the stories corresponding to that topic are grouped together although the grouped stories may actually have been placed in a wrong topic. For example, in R_1, the stories corresponding to topic t_3 are grouped together and are incorrectly identified as being relevant to t_2. We classify the errors in this case as precision errors and not as recall errors. Therefore, if even if one of the stories relevant to t_3 ($S_{401}, \ldots, S_{450}$) had been identified as being relevant to a topic other than t_2, recall would have been below 1. Similarly, for responses R_2, and R_3, the stories corresponding to t_3 and t_2 respectively are grouped together but judged relevant to the wrong topics.

To illustrate how recall is computed, we compute the recall for a new response, R_4, below. R_4 classifies $S_1, \ldots, S_{200}$ as relevant to t_1, $S_{201}, \ldots, S_{440}$ as relevant to t_2, $S_{441}, \ldots, S_{450}$ as relevant to t_3, and $S_{451}, \ldots, S_{500}$ as relevant to t_4. Therefore, $R_1 = R_2 = \ldots = R_{200} = \frac{200}{200}$ since for each of these stories, s, the system identifies 200 stories as relevant to the topic that s is actually relevant to (t_1). In addition, $R_{201} = \ldots = R_{400} = \frac{200}{200}$ since for each of these 200 stories, s, the system identifies the 200 stories relevant to the topic that s is actually relevant to (t_2). It should be noted that the denominator in this case is 200, and *not* 240, as the total number of stories deemed by **truth** as relevant to topic that each of these stories is relevant to is 200 (corresponding to topic t_2). Similarly, $R_{401} = \ldots = R_{440} = \frac{40}{50}$ since for each of these stories the system identifies 40 of the 50 stories in the topic that the story itself is actually relevant to t_3. Moreover, $R_{441} = \ldots = R_{450} = \frac{10}{50}$. Finally, $R_{451} = \ldots = R_{500} = \frac{50}{50}$. Therefore, the final recall, the average of the recalls for the 500 stories, equals

$$\frac{1}{500}\left\{200 * \frac{200}{200} + 200 * \frac{200}{200} + 40 * \frac{40}{50} + 10 * \frac{10}{50} +\right.$$

$$\left. 50 * \frac{50}{50}\right\} = \frac{200+200+32+2+50}{500} = \frac{484}{500} = 0.968.$$

5. The Weighted Method

As mentioned earlier, the weighted method assigns equal weights to all topics as opposed to equal weights for all stories. This section describes the weighted methods for both the current and the B-CUBED algorithms.

5.1. Current Algorithm

Figure 5 shows the formulae for P_{Miss}, and P_{FA} when the topics have equal weights. In the formulae, N_{Topics} equals the total number of topics. Therefore, for computing the miss rate, the formula for P_{Miss} calculates, for each topic, the percentage of relevant stories which were marked not relevant by the system. Similarly, the false alarm rate is computed by calculating, for each topic, the percentage of non relevant stories which are marked relevant by the system. Final miss and false alarm rates are computed by taking an average for the percentages obtained for each of the topics.

$$P_{\text{Miss}} = \frac{1}{N_{Topics}} \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left\{ (1 - \delta_{sys}(t,s)).\delta_{ref}(t,s) \right\} \Big/ \sum_{s \in Stories_t} \delta_{ref}(t,s) \right\}$$

$$P_{\text{FA}} = \frac{1}{N_{Topics}} \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left\{ \delta_{sys}(t,s).(1 - \delta_{ref}(t,s)) \right\} \Big/ \sum_{s \in Stories_t} (1 - \delta_{ref}(t,s)) \right\}$$

Figure 5: Equations for the Current Evaluation Algorithm – Weighted Method

For example, consider the response R_1 described earlier in the paper. The miss rate for this response equals

$$\frac{1}{4}\left\{\frac{0}{200} + \frac{0}{200} + \frac{50}{50} + \frac{0}{50}\right\} = \frac{1}{4} = 0.25.$$

Moreover, the false alarm rate for this response equals

$$\frac{1}{4}\left\{\frac{0}{300} + \frac{50}{300} + \frac{0}{450} + \frac{0}{450}\right\} = \frac{1}{24} = 0.042.$$

In comparison, the miss rate for response R_2 equals

$$\frac{1}{4}\left\{\frac{0}{200} + \frac{0}{200} + \frac{50}{50} + \frac{0}{50}\right\} = \frac{1}{4} = 0.25.$$

Finally, the false alarm rate for R_2 equals

$$\frac{1}{4}\left\{\frac{0}{300} + \frac{0}{300} + \frac{0}{450} + \frac{50}{450}\right\} = \frac{1}{36} = 0.028.$$

It should be noted that while the weighted version of the current algorithm does penalize R_1 more than R_2 for false alarm error rates, the difference is very small.

5.2. B-CUBED Algorithm

Figure 6 shows the formulae for computing Precision, and Recall using the B-CUBED algorithm when the topics have equal weights. In the formulae, NT_{sys} equals the total number of topics identified by the system in the response while NT_{ref} equals the total number of topics identified in the truth. *Topics* is the set of topics in the response. Therefore, the precision formula assigns equal weights to only those topics identified in the response while the recall formula assigns equal weights to the topics in the truth.

The formulae in Figure 6 can also be explained in terms of the equations shown in Figure 4. While the unweighted version of the B-CUBED algorithm assigns equal weights to each story when computing the final precision and recall numbers, the weighted version of the algorithm assigns equal weights to each topic when computing the final precision and recall numbers. Therefore, if a response identifies three topics, t_1, t_2, t_3, with 5 stories relevant to t_1, then the weights assigned to each of these five stories would be $\frac{1}{3} * \frac{1}{5} = \frac{1}{15}$. Similarly, if the response identified 7 stories relevant to t_2, the the weights assigned to each of them would be $\frac{1}{21}$. The final precision and recall is computed by taking this weighted average of the precision and recall numbers corresponding to each story.

For example, using the formulae in Figure 6, the precision for response R_1 described earlier in the paper equals

$$\frac{1}{3}\left\{\frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{250} * \left[200 * \frac{200}{250} + 50 * \frac{50}{250}\right] + \frac{1}{50} * 50 * \frac{50}{50}\right\} = \frac{1}{3} * \left\{1 + \frac{170}{250} + 1\right\} = \frac{67}{75} = 0.893.$$

In comparison, the precision for response R_2 equals

$$\frac{1}{3}\left\{\frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{100} * \left[50 * \frac{50}{100} + 50 * \frac{50}{100}\right]\right\} = \frac{1}{3} * \left\{1 + 1 + \frac{50}{100}\right\} = \frac{5}{6} = 0.833.$$

As explained earlier, the recall for both these responses is 1. For example, the recall for response R_1 is calculated as shown below

$$\frac{1}{4}\left\{\frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{50} * 50 * \frac{50}{50} + \frac{1}{50} * 50 * \frac{50}{50}\right\} = \frac{1}{4} * 4 = 1.$$

However, the recall for response R_4 equals

$$\frac{1}{4}\left\{\frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{200} * 200 * \frac{200}{200} + \frac{1}{50} * 40 * \frac{40}{50} + \frac{1}{50} * 10 * \frac{10}{50} + \frac{1}{50} * 50 * \frac{50}{50}\right\} = \frac{1}{4} * \left\{1 + 1 + \frac{34}{50} + 1\right\} = 0.92$$

The weighted B-CUBED algorithm computes, for each topic, the precision and recall for the set of stories corresponding to the topic irrespective of the number of documents that are relevant to the topic. In other words, precision computes, for each topic, the percentage of stories correctly identified by the response as relevant to the topic out of the total number of stories identified by the response

$$P = \frac{1}{NT_{sys}} \sum_{t \in Topics} \left\{ \frac{1}{\sum_{s \in ST} \delta_{sys}(t,s)} * \left[\sum_{s \in ST} \delta_{sys}(t,s) * \left(\frac{\left\{ \sum_{s_1 \in ST} \sum_{t_1 \in Topics} \delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1) * \delta_{sys}(t,s_1) \right\}}{\sum_{s_1 \in ST} \delta_{sys}(t,s_1)} \right) \right] \right\}$$

$$R = \frac{1}{NT_{ref}} \sum_{t \in Topics} \left\{ \frac{1}{\sum_{s \in ST} \sum_{t_1 \in Topics} \delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1)} * \left[\sum_{s \in ST} \delta_{sys}(t,s) * \left(\frac{\left\{ \sum_{s_1 \in ST} \sum_{t_1 \in Topics} \delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1) * \delta_{sys}(t,s_1) \right\}}{\sum_{s_1 \in ST} \sum_{t_1 \in Topics} \delta_{ref}(t_1,s) * \delta_{ref}(t_1,s_1)} \right) \right] \right\}$$

Figure 6: Equations for the B-CUBED Algorithm – Weighted Method

as relevant to the topic. Similarly, recall computes, for each topic, the number of stories correctly identified by the response as relevant to the topic out of the total number of stories that are actually relevant to the topic. This feature is illustrated by the example below. In this example, we proportionally reduce the size of the topics (in terms of the number of stories actually relevant to the topics) used in the previous example by a factor of 50. Therefore, the truth now assigns documents $S_1, \ldots, S_4$ to topic t_1, $S_5, \ldots, S_8$ to topic t_2, S_9 to t_3, and S_{10} to t_4. Suppose response R_5 assigns $S_1, \ldots S_4$ to t_1, $S_5, \ldots, S_9$ to t_2, and S_{10} to t_4. In other words R_5 mirrors R_1 described earlier, the only difference being the size of the topics. The precision for R_5 equals

$$\frac{1}{3}\left\{\frac{1}{4} * 4 * \frac{4}{4} + \frac{1}{5} * \left[4 * \frac{4}{5} + 1 * \frac{1}{5}\right] + \frac{1}{1} * 1 * \frac{1}{1}\right\} =$$

$$\frac{1}{3} * \left\{1 + \frac{17}{25} + 1\right\} = \frac{67}{75} = 0.893$$

which is the same as the one for R_1. Similarly, the recall for R_5 is also 1. Due to this feature of the weighted B-CUBED algorithm, we prefer the unweighted B-CUBED algorithm. However, there may be scenarios when the weighted version is preferable.

6. Conclusion

Given the build-test-improve cycle of development that is dominant in the development of natural language processing systems today, there is greater reliance on the use of standard evaluation algorithms than ever before. The type of scoring algorithm used greatly influences system development, and tuning. Therefore, it is extremely important that these scoring algorithms measure diverse aspects of the system performance so that the developers can use the collective scores to tune their systems accordingly. The B-CUBED algorithm presented in this paper is meant to enhance the current TDT tracking evaluation algorithm by providing two additional aspects: precision and recall to measure the performance of these systems.

7. References

1998. *The TDT Web Page*. NIST, http://www.nist.gov/speech/tdt3/tdt3.htm.

Bagga, Amit and Breck Baldwin, 1998. Algorithms for Scoring Coreference Chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*.