Topic Detection and Tracking with Spatio-Temporal Evidence

Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi

Department of Computer Science, University of Helsinki, Finland
{jamakkon,hahonen,salmenki}@cs.helsinki.fi,
WWW home page: http://www.cs.helsinki.fi/group/doremi

Abstract. Topic Detection and Tracking is an event-based information organization task where online news streams are monitored in order to spot new unreported events and link documents with previously detected events. The detection has proven to perform rather poorly with traditional information retrieval approaches. We present an approach that formalizes temporal expressions and augments spatial terms with ontological information and uses this data in the detection. In addition, instead using a single term vector as a document representation, we split the terms into four semantic classes and process and weigh the classes separately. The approach is motivated by experiments.

1 Introduction

Topic Detection and Tracking (TDT) is fairly recent area of information retrieval. It aims to monitor the online news stream in order to automatically spot new unreported news events (*first story detection*) and assigning documents to previously detected events (*topic tracking, cluster detection*)(see e.g. [1–3]). For example, think of an information worker or a specialist who has to deal with several incoming news-streams that report various things taking place in the world. The information worker might want to follow the course of events regarding bush fires in Australia, the development of the presidential elections in France, or just be informed if anything new takes place in Portugal or in the metal industry, for example. Given a news story, a TDT system would have to be able to attach it to any previous discussions about the event portrayed in the story – else the story would be regarded as new. The process of detecting new events has been considered difficult and the existing information retrieval methodology has had difficulties in this kind of event-based information organization [4].

We present an approach for TDT that exploits *semantic classes*, i.e., classes consisting terms that have similar meaning: locations, proper names, temporal expressions and general terms. Instead of the traditional document vector, our representation has four vectors that reside in disparate spaces. In addition, we formalize temporal expressions and provide them an interpretation on a global time-line and we evaluate the relevance of two spatial references with respect to an ontology. We outline a simple approach utilizing this kind of complex representations and compare it with single-vector methods.

This paper is organized as follows: Section 2 gives a short introduction to the previous results in TDT. The event vectors are presented in Section 3 and Section 4 deals with the comparing these vectors. Section 5 illustrates our experiments. Section 6 is a conclusion.

2 Previous Work

TDT related research begun in 1996 with DARPA funded pilot study [1]. The researchers set out to experiment the feasibility of TDT systems using existing technology. Quite soon the traditional methods for information retrieval were found more or less inadequate for online detection purposes. First story detection was characterized *queryless information retrieval* as we do not know what we are looking for, i.e., we want to detect the unexpected, *new.* Thus, query-based retrieval methods seemed insufficient [2]. The tracking task is similar to information filtering but with very few examples to work with. Since the tasks are interrelated, the poor performance in detection results in poor tracking performance. Allan, Lavrenko and Jin reduced the topic detection to topic tracking, and showed that the performance of tracking is unacceptably low for efficient first story detection. They concluded that "*effective first story detection is either impossible or requires substantially different approaches*" [4].

Furthermore, the concept of event is problematic: though it appears to be intuitively quite clear, it is difficult to establish a solid definition. Usually, it is understood as *"something happening in a certain place at a certain time"* [5]. Soon after the launching of TDT program, the scope was confined to *event detection and tracking* (e.g. [6]), but recently the focus has returned to spotting dynamic *topics* that center around a seminal events [3, 7]. However, the definition one adopts has an impact on the performance of the system [8].

The methods applied in TDT cover a good portion of the prevailing IR methods: the majority of the approaches in TDT have relied on some sort of clustering: Single-Pass Clustering [1, 2, 8] or hierarchical Group-Average Clustering [2]. Also, Hidden Markov Models [9], Rocchio [10], k-Nearest Neighbours [10], naive Bayes [11], probabilistic Expectation-Maximization models [12] and Kullback–Leibler divergence [7] have been used.

In these approaches, the documents are represented as vectors while the events are either centroids, i.e., compilations of the vectors assigned to the event, or a set of document vectors without generalization, as is the case with kNN. The terms have been weighted with tf-idf variants [4, 10], surprisingness [6], and Time Decay [1], for instance. Allan *et al.* investigated the use of named entities (NE) in the vector model [13]. Similarly, Yang *et al.* [14] extracted locations, names of individuals and organizations, time and date references, and sums of money and percentages for NE-weighting.

3 Event Vector

Making the distinction between two different air disasters or train accidents has not been easy. The terms of two documents discussing the same *kind* of event tend to converge and therefore a term vector is not able to represent the delicate distinction between documents regarding similar but not the same event [1]. However, Allan, Lavrenko and Papka suspect that only a small number of terms is adequate to make the distinction between different news events [6]. Intuitively, it would be temporal expressions, locations and names that would vary more than other terms.

A news document regarding an event reports at the barest *what* happened, *where* it happened, *when* it happened, and *who* was involved. Previous detection and tracking approaches have tried to encapsulate these facts in a single vector. In order to attain the delicate distinctions mentioned above, to avoid the problems with the term-space maintenance and still maintain robustness, we assign each of the questions a *semantic class* [8], i.e., the words that have meaning of the same type. The semantic class of LOCATIONS contains all the places mentioned in the document, and thus gives an idea, where the event took place. Similarly, TEMPORALS, i.e., the temporal expressions name a logical object, that is, a point of on a global time-line, and bind the document onto the time-axis. NAMES are proper names and tell who was involved. What happened is represented by 'normal' words which we call TERMS. These comprise nouns, adjectives and verbs.

The representation of the document using semantic classes is illustrated in Figure 1. This *event vector* comprises four sub-vectors that reside in distinct spaces due to the semantical dissimilarity. If two documents coincide in temporal expressions and locations, for example, it would suggest that they are discussing the same event. Obviously, news events are reported quite promptly, and thus the temporal similarity would be quite high for the news published on the same day. Likewise, the spatial similarity based solely on large areas, such as continents, is of course weaker than similarity based on more specific locations.

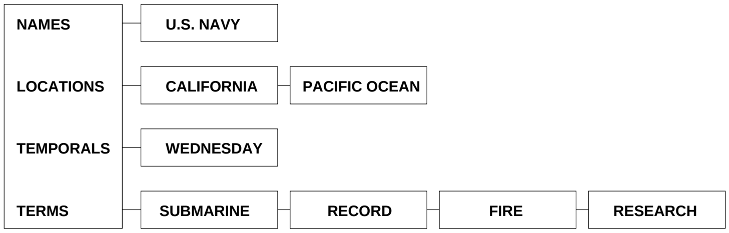

Fig. 1. An example of event vector. *"The U.S. Navy diesel research submarine that holds the world's deep-diving record caught fire in the Pacific Ocean off California on Wednesday and all 43 people aboard were rescued, the Navy said." (Washington Post, May 22, 2002)*

4 Measuring Similarity

The use of semantic classes enables us to perform the similarity comparisons class-wise, i.e., examining the corresponding sub-vectors of two event vectors at a time. This results in slight difference in the ways of determining the similarity. First, we present a general term weighting approach, which is elaborated from our previous work [8]. Then, we outline comparison of temporal and spatial references, and finally our detection and tracking algorithm.

4.1 General Term Weight

Typically, the short online news differ from detective stories in that they give away story in the first few sentences. We aim to exploit this structural feature in term weighting. Thus, we use the *ranking* of each occurrence of the term, i.e., the ordinal of the sentence in which the term takes place in measuring the importance of the term. The *rank-score* of a term t occurring m times is

$$rs(t) = \sum_{k=1}^{m} \frac{1}{2^{\ln t_k}}, \qquad (1)$$

where t_k is the ranking of the kth instance of term t. With rank-scoring, the instances of terms in the first sentence (or title) are assigned weight $\frac{1}{2^{ln 1}} = \frac{1}{1} = 1$. The rank-score decays as the ranking of the sentence grows, but the (natural) logarithm is there to modify the difference between two consecutive rankings: instances in the eighth and ninth sentences have a difference only of 0.019 $(= 0.237 - 0.218)$.

In order to determine the weight of the intersection of two documents, we calculate the ratio between the rank-score of the intersection and the rank-scores of the documents. Naturally, the informativeness of the terms themselves varies as well. Thus, loyal to the traditions of IR, we multiply the rank-scores with inverted document frequency, IDF. For example, let X and Y be sets of terms. Then their ranking-weighted similarity (RWS) equals to

$$RWS(X,Y) = \frac{\sum_{k=1}^{|X \cap Y|} rs(t_k) * IDF(t_k)}{\sum_{j=1}^{|X|} rs(t_j) + \sum_{l=1}^{|Y|} rs(t_l)} \qquad (2)$$

The intersection $|X \cap Y|$ contains all the occurrences of terms common to both documents. Thus, if word *'airport'* occurs twice in X and once in Y, there are three occurrences in the intersection. Therefore, the weight of the intersection equals to 1, if the two documents are identical, and 0 if the documents have no common terms.

4.2 Temporal Similarity

Temporal expressions often convey their information implicitly. This means that by examining the surface forms is seldom of any avail. For example, finding the expression 'last Monday' in two documents tells little of their similarity, since the referent of the expression changes with respect to the moment of utterance. We construct automata for temporal expression pattern recognition similarly to [15]. The found patterns do not make sense without augmented information, and thus we *canonize* the expressions with a formalized *calendar* [16] and a set of *shift* and *span* operations [17]. As a result, we provide each recognized expression with a semantical interpretation as an interval on a global time-line $\mathcal{T}$ with respect to the publication date-stamp.

In our approach, the temporal similarity of two documents is a result of a pair-wise comparison of the expressions: each start-end pair of one document is compared to each of the start-end pairs of the other. Krippendorff has conducted various investigations with intervals [18] and motivated by his work we propose a cross-tabulation illustrated in Figure 2. It shows intervals of two sets $A = \{A_1, A_2, A_3\}$ and $B = \{B_1, B_2, B_3, B_4\}$ on time-axis t. The diagonal represents the synchronous points between the two time-axis. The shaded areas correspond to the overlapping intervals. For example, A_3 and B_4 have matching starting point on the time-axis, but mismatching end points. Thus, B_4 covers A_3 only partially.

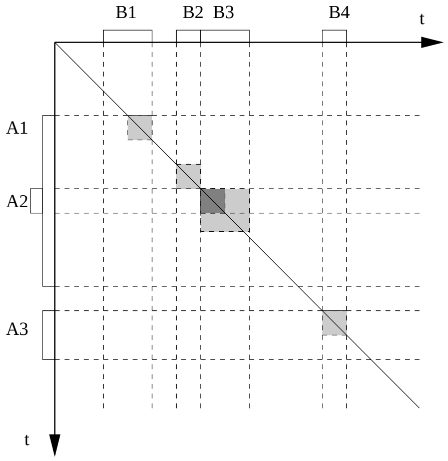

Fig. 2. A cross-tabulation of two sets of intervals A and B.

If the two sets contain the same intervals, they cover each other completely. In such case, all of the intervals would be shaded completely along the diagonal in Figure 2. In case there are disparate intervals, the larger intervals provide weaker coverage than shorter ones. As an example, consider comparing a day and a year versus a day and a weekend.

Galton lists 13 possible relations for two intervals [19]. In Table 4.2, we are not concerned, whether A is **before** B or vice versa, and hence the number of relations is decreased down to seven. We want to take these relations into

Table 1. The possible relations of two intervals. Note that the first six relations also have the converse.

$[t_i, t_j]$ is **before** $[t_k, t_l]$	if $t_j < t_k$
$[t_i, t_j]$ **meets** $[t_k, t_l]$	if $t_j = t_k$
$[t_i, t_j]$ **overlaps** $[t_k, t_l]$	if $t_i < t_k < t_j < t_l$
$[t_i, t_j]$ **begins** $[t_k, t_l]$	if $t_i = t_k \wedge t_j < t_l$
$[t_i, t_j]$ **falls within** $[t_k, t_l]$	if $t_i < t_k \wedge t_j < t_l$
$[t_i, t_j]$ **finishes** $[t_k, t_l]$	if $t_i < t_k \wedge t_j = t_l$
$[t_i, t_j]$ **equals** $[t_k, t_l]$	if $t_i = t_k \wedge t_j = t_l$

account while comparing the temporal evidence of two documents. The more the intervals overlap each other with respect to their lengths, the higher the similarity. We employ a simple weight function $\mu_t : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ such that

$$\mu_t([t_i, t_j], [t_k, t_l]) = \frac{2\,\Delta([t_i, t_j] \cap [t_k, t_l])}{\Delta(t_i, t_j) + \Delta(t_k, t_l)}, \tag{3}$$

where $\Delta : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$, $\Delta(t_i, t_i) = 1$ is the duration (in days) of the given interval. The weight function results in 1 if the expressions are an exact match and 0 if the expressions are distinct. All of the relations presented above are contained within the μ_t-function, since they can be represented in terms of the intersection.

In Figure 2, the intersections $A_3 \cap B_4$ and $A_2 \cap B_3$ would result in higher μ_t-value than the any of the intersections $A_1 \cap B_1$, $A_1 \cap B_3$, and $A_1 \cap B_2$, because the sizes of $A_3 \cap B_4$ and $A_2 \cap B_3$ are closer to the sizes of the union of the intervals, i.e., $|A_3 \cup B_4|$ and $|A_2 \cup B_3|$, and thus there is less uncovered area.

In practice, the pair-wise μ_t-weights are calculated in what we call a *cover matrix* illustrated in Table 2. The coverage of an interval $T_{i,j}$ is calculated by choosing the maximum $v_{i,j}$ of the weights for that term. If an interval $T_{1,i}$ is covered with an interval $T_{2,j}$ of equal weight, the maximum value is $v_{1,i} = 1$. On the contrary, if it is not covered at all, the maximum value yields $v_{1,j} = 0$. In cases of partial or weak cover the value varies in $(0, 1)$ depending on the sizes of the intervals.

Table 2. A cover matrix. The maximum coverage for the interval $T_{1,1}$ would yield $v_{1,1} = \max_{j \leq m}(\mu_t(T_{1,1}, T_{2,j}))$.

	$T_{2,1}$	$\ldots$	$T_{2,m}$	max
$T_{1,1}$	$\mu_t(T_{1,1}, T_{2,1})$	$\ldots$	$\mu_t(T_{1,1}, T_{2,m})$	$v_{1,1}$
$\vdots$	$\vdots$		$\vdots$	
$T_{1,n}$	$\mu_t(T_{1,n}, T_{2,1})$	$\ldots$	$\mu_t(T_{1,n}, T_{2,m})$	$v_{1,n}$
max	$v_{2,1}$		$v_{2,m}$	

The total *coverage* of the two sets of intervals is the sum of all the maximum values $v_{i,j}$ divided by the number of intervals. Let T_1 and T_2 be sets of intervals such that T_1 contains n intervals and T_2 contains m intervals. The coverage of the intervals is

$$cover_t(T_1, T_2) = \frac{\sum_{i=1}^{n} v_{1,i} + \sum_{j=1}^{m} v_{2,j}}{n+m}. \tag{4}$$

Because $\mu_t = 1$ stands for the perfect match, $v_{i,j} \in [0,1]$ and, since $cover_t(T_1, T_2)$ is really an average of the maximums, also $cover_t(T_1, T_2) \in [0,1]$

We want to weight the temporal expressions with respect to the their ranking-weighted similarity of Equation 2, but without the IDF-weight. Thus the temporal similarity of documents X and Y yields

$$sim_t(X, Y) = cover_t(X_t, Y_t) * RWS'(X_t, Y_t) \tag{5}$$

where X_t and Y_t are the temporal expressions in X and Y, respectively, and $RWS'(X_t, Y_t)$ is the ranking score without the IDF-value.

4.3 Spatial Similarity

The introduction of a geographical ontology enables measuring similarity of the spatial references on a finer scale than just binary decision match–mismatch. For example, when reporting floods in Siberia, the terms such as Russia, Lena, Vilyuy, Lensk and Yakutsk have nothing in common in the surface forms, but their geographical proximity and relevance can be understood by the virtue of an ontology. In other words, we tie each spatial expression to a global structure and thus provide it with a meaning that relates to other spatial expressions.

We employ a 5-level hierarchy in our knowledge of the world as portrayed in Table 3. The levels involved depend on the type of the location. As to land, the levels are continent, region, country, administrative region (e.g., province, state, commune, municipality, municipio, gemeente, kommun), and city. In addition to administrative region, level 4 can also be mountains, seas, lakes and (larger) rivers that include or connect to mountain peaks and (smaller) rivers.

Table 3. An example of ontology.

Location	Type	Level 1	Level 2	Level 3	Level 4	Level 5
Delft	city	Europe	W.Europe	Netherlands	Zuid-Holland	Delft
Europe	continent	Europe	–	–	–	–
Haag	city	Europe	W.Europe	Netherlands	Zuid-Holland	Haag
Main	river	Europe	W.Europe	Germany	Rhine	Main
Netherlands	country	Europe	W.Europe	Netherlands	–	–
North Sea	sea	Atlantic	North Sea	–	–	–
Rhine	river	Europe	W.Europe	Switzerland, Germany, France, Netherlands	North Sea	Rhine

Figure 3 shows a simplified taxonomy containing a number of places. Each node in the tree stands for a location. In case we want to measure the similarity of two such locations, we compare the length of the common path to the sum of the lengths of the paths to the elements, and hence the spatial similarity μ_s of two spatial terms l_1 and l_2 yields

$$\mu_s(l_1, l_2) = \frac{(level(l_1 \cap l_2))}{(level(l_1) + level(l_2))} \tag{6}$$

In case of identity, we assign $\mu_s(l_1, l_1) = 1$. Now, comparing France and Germany would result in $1/(2+2) = 1/4$ since the length of the common path (Europe) is 1 and the length of path to both France and Germany equals to 2. Similarly, comparing China and Paris would result in $0/(2+3) = 0$. Paris and France have similarity of $2/(2+3) = 2/5$.

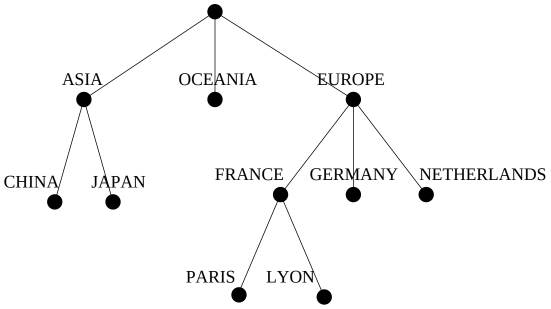

Fig. 3. A simplified ontological taxonomy.

Since all the spatial references of one document are to be compared with all of the spatial references of another, we employ the cover matrix presented in Section 4.2. For each term we choose only the maximum similarity, and let the average of maximums stand for the spatial similarity of two documents analogously to temporal coverage. Let L_1 and L_2 be sets of spatial terms such that L_1 contains n terms and L_2 contains m terms, respectively. The spatial coverage is defined as follows

$$cover_s(L_1, L_2) = \frac{\sum_{i=1}^{n} v_{1,i} + \sum_{j=1}^{m} v_{2,j}}{n+m}. \tag{7}$$

Analogously to Equation 5, although here we employ IDF, the spatial similarity of documents X and Y is

$$sim_s(X, Y) = cover_s(X_s, Y_s) * RWS(X_s, Y_s) \tag{8}$$

where X_s and Y_s are the spatial references in X and Y.

4.4 TDT Algorithm

As stated in Section 2, the detection of first stories relies on the tracking. In other words, if a document is not found sufficiently similar to any of the previously detected ones, it is considered a first story. This kind of method is called *single-pass clustering* [20], as the cluster of a new data point is resolved in a single run. We employ two kinds of approaches: one using Kullback-Leibler divergence and another of heuristic kind.

Skew Divergence Kullback-Leibler divergence measures the distance between two probability mass functions. It has been used with relevance models in TDT with some success [7]. We adopt it in a different manner: in order to determine the relative significance of the evidence of each semantic class, we build a model for similarity, m_{yes}, and a model for dissimilarity, m_{no}. The models are average distributions of pair-wise comparisons in the training material. The underlying assumption is that the model for similarity m_{yes} has higher values in each of the semantic classes than those of the model of dissimilarity m_{no}. Therefore, when comparing two documents that discuss the same event, the distribution of the class-wise comparison should be closer to the model m_{yes} than to the model m_{no}

We utilize the Kullback-Leibler divergence to measure the distance to both of the models to see whether the output of the comparison is closer to the average distribution between two documents on the same or different event. Thus, we write

$$D(m||r) = \sum_{c} m(c)(\log m(c) - \log r(c)) \tag{9}$$

where c is a semantic class, m is the model, and r is the distribution of the similarity per semantic class. Since the results of the semantic class comparisons do not necessarily yield a probability distribution, we need to tackle the zero values. Instead of smoothing, we adopt the Skew Divergence [21],

$$s_{\alpha}(r, m) = D(m||\alpha r + (1 - \alpha)m) \tag{10}$$

where $\alpha \in [0, 1]$. Now, the algorithm described in Figure 4 uses the ratio of the Skew Divergence with the similarity and dissimilarity models, i.e.,

$$\frac{s_{\alpha}(r, m_{yes})}{s_{\alpha}(r, m_{no})}, \tag{11}$$

in determining to which the comparison result r is closer to. The suitable threshold value θ is obtained by empirical experiments with the training data.

The algorithm proceeds as follows: Initially, the set of events is empty as we start processing the incoming documents one by one. The document vector v has a sub-vector v_c for each semantic class c. The document vector is then compared to each of the found events, and the results from the class-wise comparisons are stored in distribution $dist$. If the maximum of Equation 11 exceeds the threshold θ, the vector of the resulting event is updated (line 16). Otherwise, the document is considered a first story and is added to the found events.

```
1   $found \leftarrow ()$;
2   for each new document $d$
3       $v \leftarrow$ buildVector($d$);
4       $max \leftarrow 0$; $event \leftarrow ()$;
5       for each found $e$
6           $dist \leftarrow ()$;
7           for each semantic class $c$
8               add($sim_c(v_c, e_c), dist$);
9           end;
10          if ( $\mathrm{s}_\alpha(dist, \mathrm{m}_{yes}) /\ \mathrm{s}_\alpha(dist, \mathrm{m}_{no}) > max$ )
11            then $max \leftarrow \mathrm{s}_\alpha(\mathrm{dist}, \mathrm{m}_{yes}) /\ \mathrm{s}_\alpha(\mathrm{dist}, \mathrm{m}_{no})$);
12                $event \leftarrow e$;
13          fi;
14      end;
15      if ( max $> \theta$ )
16          then update($event$, $v$);
17      else add($v$, $found$);
18      fi;
19  end;
```

Fig. 4. A single-pass clustering algorithm using Skew Divergence.

Heuristic Thresholding Another approach is to assign heuristically found weights to semantic classes. The difference to the algorithm of Figure 4 is that on lines 10 and 11 there is a sum of the similarity scores of the semantic classes,

$$\sum_{c \in C} \beta_c * sim_c(v_c, e_c), \tag{12}$$

instead of Equation 11. The β_c reflects the importance of semantic class c with respect to the others, for we do not consider semantic classes equally important. That is, we multiply the similarity of the LOCATIONS, NAMES, TERMS and TEMPORAL with $\beta_{locations} = 2.0, \beta_{names} = 2.0, \beta_{terms} = 0.8$ and $\beta_{temporal} = 1.0$, respectively. TEMPORAL evidence is the least important since it tends to be high for the documents published on the same day. On the average, TERMS co-occur more frequently than NAMES and LOCATIONS, and hence the latter two have higher weights. A proper optimization would be an obvious improvement. However, the optimization criteria would be rather tricky, because the evaluation of a TDT is system is not straight-forward.

We also reward for having positive values in any three of the classes NAMES, LOCATIONS, TEMPORAL and TERMS, and especially if there non-zero values in all of them. On the contrary, we do not want to determine two documents similar based only on LOCATIONS, TEMPORALS or NAMES, and therefore we punish for the absence of evidence of TERMS. In practice, rewarding means multiplying with 1.5 and punishing by 0.5.

5 Experiments

5.1 Corpus

Our corpus consists of 10384 Finnish online news documents from April 1st 2001 to December 31st, 2001. We have manually assigned 5807 documents to events. The training material consisting of 1918 documents yields 79 events and the testing material comprises 3909 documents with 85 events. The events in the testing set vary from the Siberian floods and the prolonged doctors' strike in Finland to the first space tourist, the presidental elections in Peru and the riots of June 2001 in Gothenburg, Sweden.

We employ Connexor's [1] functional dependency grammar based parser in extracting TERMS, i.e., nouns, adjectives and verbs. The details of our approach to recognizing and resolving temporal expressions are reported in [17]. In extracting LOCATIONS and NAMES we rely on Connexor's Named Entity recognizer. Table 4 describes the average document in the corpus. There are less than 5 instances of LOCATIONS and over 6 instances of NAMES in each document on the average. The portion of TERMS is considerably larger than that of any of the other classes.

Table 4. Test c orpus statistics: $Exp(X)$ is the expectation, $Var(X)$ the variance and $Std(X)$ the standard deviation of the size X of the given semantic class.

semantic class	$Exp(X)$	$Var(X)$	$Std(X)$
LOCATIONS	4.460	16.698	4.086
NAMES	6.541	37.629	6.134
TERMS	56.363	576.363	24.008
TEMPORALS	2.669	5.013	2.239
total words	105.578	1773.370	42.285

In addition, we have manually classified the testing documents to 17 categories that form the first level of the International Press and Telecommunications Council (IPTC) taxonomy [2]. The distribution of the classes is illustrated in Figure 5. On the average, a document is assigned to 1.46 categories. The largest classes are number 4, *economy, business and finance*, and number 11, *politics*.

This classification has been done in order to decrease the number of pairwise comparisons. Although our corpus at present does not encourage to build a classifier, the reported performance of automatic text categorization, however, makes the use of the classes highly feasible. There are four documents in the test set that are classified outside of the class of the first story, and they cannot be correctly tracked. In other words, these four documents do not have mutual categories with the rest of the documents dealing with the same events.

[1] http://www.connexor.com

[2] http://www.iptc.org

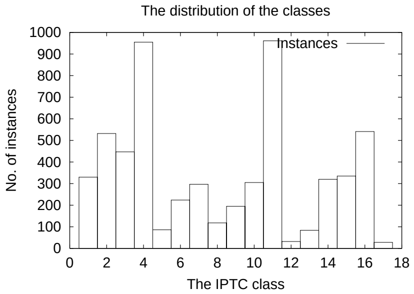

Fig. 5. The distribution of IPTC classes in the test corpus.

The contents of our ontology is listed in Table 5. The data is based on material provided by Statistics Finland [3]. Since the corpus contains a good number of domestic events, we have added another ontology from the same source in addition to the global one. The domestic locations contain all the counties, provinces and communes of Finland.

5.2 Detection and Tracking Results

We have made the following assumptions: The documents that do not have an event assigned to them in the corpus do not count as first stories. In addition, if two documents that are not assigned to any event are found to discuss the same event, it does not affect the results. These unlabeled documents interfere with the tracking, if they are assigned to some event or if some labeled document is found similar to them.

The methods were evaluated with precision, recall, and their combination F1-measure. The evaluation measures comply with the following formulas:

$$\text{Precision} = P = \frac{relevants\ found}{all\ found}$$
$$\text{Recall} = R = \frac{relevants\ found}{all\ relevant}$$
$$\text{F1-measure} = F1 = \frac{2PR}{P+R}$$

[3] `http://www.stat.fi`

Table 5. Ontology statistics.

type		type	
continents	6	mountain peaks	269
regions	23	mountains	116
countries	270	rivers	369
administrative districts	1422	domestic locations	576
cities	4116	oceans/seas	77
deserts	35	lakes	276

An event is represented by a centroid, or actually the average of the first and the last document assigned to an event.

We ran experiments with Skew Divergence and Heuristic Thresholding. In addition, in order to provide a baseline, we ran test also with Cosine coefficient [20], with and without the semantic classes. Table 5.2 shows the results of the experiments. In order to compare the methods, we combined the F1-measures to indicate overall efficiency of each method. The average is listed on the right. The overall F1-measure was maximized to obtain the results. Each row is produced by one threshold value, i.e., the same threshold is used in both the tracking and the detection. The considerable difference between precision and recall shows the difficulty of optimizing both tasks at the same time. Because the tasks are so interrelated, it is hard to come up with a good optimization criteria.

Secondly, the performance of Skew Divergence seems very poor. Either there is something wrong with the model, or four variables is not enough for measuring divergence. Presumably, this kind of modeling requires larger masses of data and variables. However, the result does not contradict those reported in [22]: Kullback-Leibler, though applied in different way, performed consistently worse precision and recall than Cosine.

Table 6. The results of detection and tracking

	Detection			Tracking			
method	P	R	$F1_D$	P	R	$F1_T$	$\frac{F1_D+F1_T}{2}$
Cosine	0.473	0.237	0.315	0.214	0.766	0.334	0.325
Cosine (SC)	0.531	0.294	0.379	0.286	0.500	0.363	0.371
Skew Divergence	0.400	0.190	0.258	0.207	0.545	0.300	0.279
Heuristic	0.551	0.905	0.685	0.688	0.450	0.544	0.620

Another striking observation is the high performance of the heuristic approach. Simple rules based on intuition and observations outperform all of the other methods by far. The high recall in detection is probably due to the lower precision: since there are more documents considered first stories, there are more correct ones. A decent precision in tracking also helps.

Uniformly through out the results, there seems to be a connection between the detection precision and the tracking recall as well as between the detection recall and the tracking precision. A high value in one results in a high value in the other.

In all, the results, though modest, are at least not considerably worse than those reported by Papka [5], for example. They are still less than what Allan *et al.* would call acceptable.

6 Conclusions

We have presented a topic detection and tracking approach that employs semantic classes in event representation. We identified four classes, places, names, temporal expressions and general terms, and ran the comparisons of two documents class-wise. The approach relies on heavy use of NLP techniques.

We have also presented a method to compare temporal and spatial information in the context of TDT. The method enables the comparison of two relevant terms that differ in the surface forms.

We used a divergence of models and heuristic approach in the detection, and provided results of plain and simple cosine coefficient as a baseline.

In the future, we will obviously concentrate on developing the models more accurate, that is, finding ways in which to represent the yes- and no-distributions with less noise. We will also make efforts to build the heuristic approach a solid theoretical background. Also, we will run the experiments on the Linguistic Data Consortium's TDT data in order to have results that are fully comparable with the previous work.

References

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study final report. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop. (1998)
2. Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B.T., Liu, X.: Learning approaches for detecting and tracking news events. IEEE Intelligent Systems Special Issue on Applications of Intelligent Information Retrieval **14** (1999) 32 – 43
3. Allan, J., ed.: Topic Detection and Tracking – Event-based Information Organization. Kluwer Academic Publishers (2002)
4. Allan, J., Lavrenko, V., Jin, H.: First story detection in TDT is hard. In: Proc. 9th Conference on Information Knowledge Management CIKM, McClean, VA USA (2000) 374–381
5. Papka, R., Allan, J.: On-line new event detection using single-pass clustering. Technical Report IR–123, Department of Computer Science, University of Massachusetts (1998)
6. Allan, J., Lavrenko, V., Papka, R.: Event tracking. Technical Report IR – 128, Department of Computer Science, University of Massachusetts (1998)
7. Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., Thomas, S.: Relevance models for topic detection and tracking. In: Proc. Human Language Technology Conference (HLT). (2002)
8. Makkonen, J., Ahonen-Myka, H., Salmenkivi, M.: Applying semantic classes in event detection and tracking. In: Proc. International Conference on Natural Language Processing (ICON'02), Mumbai, India (2002)
9. van Mulbregt, P., Carp, I., Gillick, L., Lowe, S., Yamron, J.: Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In: Proc. 5th Intl. Conference on Spoken Language Processing (ICSLP'98). (1998)
10. Yang, Y., Ault, T., Pierce, T., Lattimer, C.: Improving text categorization methods for event detection. In: Proc. ACM SIGIR. (2000) 65–72

11. Seymore, K., Rosenfeld, R.: Large-scale topic detection and language model adaptation. Technical report, School of Computer Science, Carnegie Mellon University (1997)
12. Baker, L.D., Hofmann, T., McCallum, A., Yang, Y.: A hierarchical probabilistic model for novelty detection in text. unpublished manuscript (1999)
13. Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., Caputo, D.: Topic-based novelty detection. Technical Report Summer Workshop Final Report, Center for Language and Speech Processing, Johns Hopkins University (1999)
14. Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned novelty detection. In: Proc. ACM SIGKDD (to appear), Edmonton, Canada (2002)
15. Schilder, F., Habel, C.: From temporal expressions to temporal information: Semantic tagging of news messages. In: Proc. ACL-2001 Workshop on Temporal and Spatial Information Processing. (2001) 65–72
16. Goralwalla, I.A., Leontiev, Y., Özsu, M.T., Szafron, D., Combi, C.: Temporal granularity: Completing the puzzle. Journal of Intelligent Information Systems **16** (2001) 41–63
17. Makkonen, J., Ahonen-Myka, H.: Extraction and comparison of temporal evidence in identifying news events. Unpublished manuscript
18. Krippendorff, K.: On the reliability of unitizing continuous data. In Marsden, P.V., ed.: Sociological Methodology. Blackwell (1995) 47–76
19. Galton, A.: Time and change for AI. In Gabbay, M., Hogger, C.J., Robinson, J.A., eds.: Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4, Epistemic and Temporal Reasoning. Oxford University Press (1995) 175–240
20. van Rijsbergen, C.J.: Information Retrieval. 2nd edn. Butterworths (1980)
21. Lee, L.: On the effectiveness of the skew divergence for statistical language analysis. In: Arficial Intelligence and Statistics. (2001) 65–72
22. Allan, J., Lavrenko, V., Swan, R.: Explorations within topic tracking and detection. In Allan, J., ed.: Topic Detection and Tracking – Event-based Information Organization. Kluwer Academic Publisher (2002) 197–224