Simple Semantics in Topic Detection and Tracking

Juha Makkonen, Helena Anonen-Myka, and Marko Salmenkivi
Introduction

- Topic Detection and Tracking (TDT) focuses on organizing news documents
- Split documents into stories, spotting new stories, tracking development of an event, and grouping together stories describing the same event
- A TDT system runs on-line without knowledge of incoming stories
- Short duration events cause changing vocabulary
• Use *semantic classes*, groups consisting of terms that have similar meaning: location, proper names, temporal expressions, and general terms

• Similarity metric is applied class-wise: compare names in one document with names in the other, the locations in one document with locations in the other, etc.

• Allows a semantic similarity between terms rather than binary string matching

• Results in a vector of similarity measures, which is combined via weighted sum to produce a yes/no decision
Topic Detection and Tracking

- Compilation of on-line news and transcribed broadcasts from one or more sources and one or more languages
- TDT consists of five tasks:
  1. Topic tracking monitors news streams for stories discussing given target topic
  2. First story detection makes binary decisions on whether a document discusses a new, previously unreported topic
  3. Topic detection forms topic-based clusters
  4. Link detection determines whether two documents discuss the same topic
  5. Story segmentation finds boundaries for cohesive text fragments
- TDT presents unique challenges: on-line, few assumptions, small number of documents, changing vocabulary
Definitions

• An event is an unique thing that happens at some specific time and place
  • Definition neglects events with either long timelines, escalating directions, or lack of tight spatio-temporal constraints

• A topic is an event or activity, along with all related events or activities
  • A topic is a set of documents that related strongly to each other via a seminal event
Document Representation

- Four types of terms: locations, temporal expressions, names, and general terms
- Introduces simple semantics since all terms in a given type are compared
Event Vector

- **Semantic classes** are assigned to basic questions in news article: who, what, when, where
  - Called NAMES, TERMS, TEMPORALS, and LOCATIONS
- An *event vector* is formed by combining multiple semantic classes
An example event vector for AP news article starting "RAMALLAH, West Bank — Palestinian leader Yassar Arafat appointed his longtime deputy Mahmoud Abbas as prime minister Wednesday..."
Comparing Event Vectors

- Comparison is done class-wise, i.e., via corresponding sub-vectors of two event representations.
- Similarity metric can be different for each class.
  - Use a weighted sum of the similarity measures for final binary decision.
- Results in a vector in $\mathbf{v} = \{v_1, v_2, v_3, v_4\} \in \mathbb{R}^4$. 
Similarity for NAMES and TERMS

- Use the term-frequency inverted document frequency
- Let $T = \{t_1, t_2, \ldots, t_n\}$ denote the terms, $D = \{d_1, d_2, \ldots, t_m\}$ denote the documents. Then, the weight $w : T \times D \to \mathbb{R}$ is defined as:

$$w(t, d) = f(t, d) \cdot \log \left( \frac{|D|}{g(t)} \right),$$

where $f : T \times D \to \mathbb{N}$ represents the number of occurrences of term $t$ in document $d$, $|D|$ is the total number of documents, and $g : T \to \mathbb{N}$ is number of documents in which term $t$ occurs (i.e., the document frequency of term $t$).
- The similarity of two sub-vectors $X_k$ and $Y_k$ of semantic class $k$ is based on the cosine of the two:

$$\sigma(X_k, Y_k) = \frac{\sum_{i=1}^{|k|} w(t_i, X_k) \cdot w(t_i, Y_k)}{\sqrt{\sum_{i=1}^{|k|} w(t_i, X_k)^2} \cdot \sqrt{\sum_{i=1}^{|k|} w(t_i, Y_k)^2}}$$

where $|k|$ is the number of terms in semantic class $k$. 
Similarity for TEMPORALS

- Time intervals are mapped to a global calendar that defines a time-line and unit conversion.
- Temporal similarity is based on comparison of intervals of each document. Let $T$ be the global timeline, $x \subseteq T$ be a time interval with start- and end-points, $x_s$ and $x_e$. Similarity between two intervals is

$$\mu_t(x, y) = \frac{2 \Delta([x_s, x_e] \cap [y_s, y_e])}{\Delta(x_s, x_e) + \Delta(y_s, y_e)}$$

where $\Delta$ is the duration of the interval in days.

- For each pair of intervals from TEMPORAL vectors $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, determine the maximum value. The similarity is the average of all these maxima, i.e.,

$$\sigma_s(X, Y) = \frac{\sum_{i=1}^{n} \max(\mu_s(x_i, Y)) + \sum_{j=1}^{m} \max(\mu_s(X, y_j))}{m + n}$$
Similarity for LOCATIONS

- Locations are split into a five-level hierarchy
  - Continent, region, country, administrative region, and city
  - Administrative region can be replaced by mountain, seas, lakes, or river
  - Represented by a tree
- Similarity between two locations, \( x \) and \( y \) is based on the length of the common path:
  \[
  \mu_s(x, y) = \frac{\lambda(x \cap y)}{\lambda(x) + \lambda(y)}
  \]
  where \( \lambda(x) \) is the length of the path from the root to the element \( x \).
- The spatial similarity between two LOCATION vectors
  \( X = \{x_1, x_2, \ldots, x_n\} \) and \( Y = \{y_1, y_2, \ldots, y_m\} \) is
  \[
  \sigma_s(X, Y) = \frac{\sum_{i=1}^{n} \max (\mu_s(x_i, Y)) + \sum_{j=1}^{m} \max (\mu_s(X, y_j))}{m + n}
  \]
Class-wise comparison of two event vectors produces results in a vector \( \mathbf{v} = \{v_1, v_2, v_3, v_4\} \in \mathbb{R}^4 \).

Similarity is based on a weighted linear sum of class-wise similarity: \( \langle \mathbf{w}, \mathbf{v} \rangle \).

Simplest algorithm uses a hyper-plane: \( \psi(\mathbf{v}) = \langle \mathbf{w}, \mathbf{v} \rangle + b \), and a perceptron to learn \( \mathbf{w} \) and \( b \).

Data is typically not linearly separable, so, transform \( \mathbf{v} \) to higher dimensional space, and use a perceptron to learn a hyper-plane there:

- Define \( \phi : \mathbb{R}^4 \to \mathbb{R}^{15} \) that expands \( \mathbf{v} \) into its powerset.
- Then hyper-plane is \( \psi(\mathbf{v}) = \langle \mathbf{w}', \phi(\mathbf{v}) \rangle + b \).
Topic Tracking Algorithm

topic ← buildVector()
For each new document d
    doc ← buildVector(d)
    v ← (), decision ← ()
    For each semantic class
        v[c] ← sim_c(doc_c, topic_c)
    If ⟨w', φ(v)⟩ + b ≥ 0
        decision = 'YES'
    else
        decision = 'NO'
First Story Detection Algorithm

topics ← (); decision ← ()
For each new document \(d\)
    doc ← buildVector(d)
    max ← 0; max_topic ← 0
For each topic
    For each semantic class
        \(v[c] \leftarrow sim_c(doc_c, topic_c)\)
        If (\(\langle w', \phi(v) \rangle + b \geq max\))
            max ← \(\langle w', \phi(v) \rangle + b\)
            max_topic ← topic
        If (max < \(\theta\))
            decision[d] ← 'first-story'
        else
            decision[d] ← max_topic
    add(topics, doc)
Experiments

- Text corpus contains 60,000 documents from two on-line newspapers, two TV broadcasts, and two radio broadcasts
- Automatic term extraction combined with automata and gazetteer to improve performance
## Topic Tracking Results

| Method          | $C_{det}$ | $(C_{det})_{norm}$ | $P_{miss}$ | $P_{fa}$ | $p$    | $r$    | $F_1$  |
|-----------------|-----------|--------------------|------------|----------|--------|--------|--------|
| Cosine          | 0.0058    | 0.0720             | 0.0100     | 0.0470   | 0.2361 | 0.7900 | 0.2927 |
| Weighted Sum    | 0.0471    | 0.5214             | 0.1818     | 0.0668   | 0.1646 | 0.8181 | 0.2741 |

**Table:** Using $(C_{det})_{norm}$

| Method          | $C_{det}$ | $(C_{det})_{norm}$ | $P_{miss}$ | $P_{fa}$ | $p$    | $r$    | $F_1$  |
|-----------------|-----------|--------------------|------------|----------|--------|--------|--------|
| Cosine          | 0.0524    | 0.6553             | 0.2582     | 0.0097   | 0.5297 | 0.7481 | 0.5481 |
| Weighted Sum    | 0.0849    | 1.0621             | 0.4242     | 0.0015   | 0.8636 | 0.5758 | 0.6910 |

**Table:** Using $F_1$
## First-Story Detection Results

| Method          | $C_{det}$ | $(C_{det})_{norm}$ | $P_{miss}$ | $P_{fa}$ | $p$  | $r$  | $F_1$  |
|-----------------|-----------|--------------------|------------|----------|------|------|--------|
| Cosine          | 0.0033    | 0.0414             | 0.0000     | 0.0414   | 0.4583| 1.0000| 0.6386 |
| Weighted Sum    | 0.0036    | 0.0446             | 0.0000     | 0.0446   | 0.4400| 1.0000| 0.6111 |

**Table:** Using $(C_{det})_{norm}$

| Method          | $C_{det}$ | $(C_{det})_{norm}$ | $P_{miss}$ | $P_{fa}$ | $p$  | $r$  | $F_1$  |
|-----------------|-----------|--------------------|------------|----------|------|------|--------|
| Cosine          | 0.0381    | 0.4768             | 0.1818     | 0.0223   | 0.5625| 0.8181| 0.6667 |
| Weighted Sum    | 0.0558    | 0.6977             | 0.2727     | 0.0159   | 0.6154| 0.7272| 0.6667 |

**Table:** Using $F_1$
Discussion

• In topic tracking, performance degrades due to lack of vagueness factor
  • For example, matching terms Asia and Washington produce the same similarity score, but does not account for indefiniteness of the terms.

• Including \textit{a posteriori} approaches that examine all the data and the labels might improve performance
Conclusions

• Paper presents a topic detection and tracking algorithm based on semantic classes
• Comparison is class-wise
• Created geographical and temporal ontologies
• Semantic augmentation degraded performance, especially in topic tracking
  • Partially due to inadequate spatial and temporal similarity function