Unsupervised Topic Discovery

Richard Schwartz (schwartz@bbn.com**)**
Sreenivasa Sista (ssista@bbn.com**)**
Timothy Leek (tleek@bbn.com**)**

BBN Technologies

Abstract

This white paper describes a new problem, which is to determine a set of topics or subjects automatically from a corpus. The result is a large number of topics, each with meaningful names. Each of the documents in the training corpus is assigned several of these topics. Finally, using the topic classification algorithms we have previously developed in OnTopic™, we estimate topic models from this corpus to use in topic classification for new documents from the same language and domain.

What are Topics?

There are many meta-definitions for topics, so it's worth defining what we mean. By "topics", we mean subjects that can be used to categorize a document, much as used by Primary Source Media or by Reuters. For example, a story about the Oklahoma Bombing might be labeled with topics like "Bombings", "Terrorism", "Oklahoma", "Deaths and Injuries". Each document is expected to have many topics assigned to it. The set of topics would usually be in the thousands. The topics cannot usually be organized into a strict tree. For example, the somewhat narrow topic "Labor Unions" should go under both "Economics" and "Politics".

Topic Categorization Model

There are many methods for determining the topics in a document. The OnTopic™ system at BBN uses a Hidden Markov Model (HMM) to model multiple topics in documents explicitly [1]. The model is pictured in Figure 1. We assume (make believe) that the story is generated by this model. According to this model, when an author decides to write a story, the first thing he does is pick a set of topics. The topics are chosen according to the prior distribution for topics. There is a HMM with one state for each of the chosen topics, plus an additional state for the General Language topic. Each of the states has a distribution for the language about that topic. In the simplest case, this language model is a unigram distribution on words. But the state could also contain a higher order ngram language model for word sequences for that topic. According to the model, the author writes the story one word at a time. Before choosing each word, he first chooses which of the topics that word will be about. This is chosen according to the probability of each topic, given the set of topics in the story. Once the topic is chosen, the author chooses a word from the corresponding topic state according to the distribution of words for the topic. Then the author must choose the topic for the next word, and so on until the story is finished. Note that the majority of the words in a story are typically

generated by the General Language state. (If an ngram model is used, the process is slightly more complicated, but the same principle can be used.)

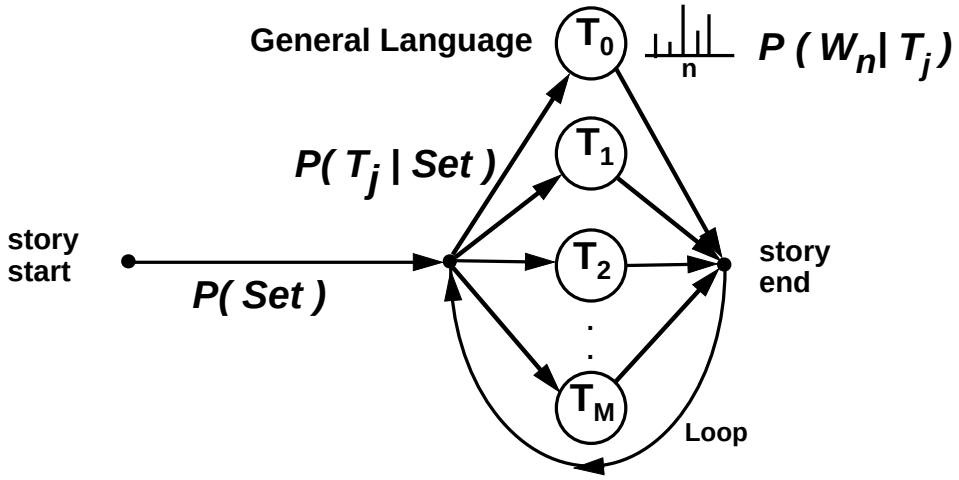

Figure 1: HMM for how a story is generated from its topics.

Typically, we are not given the word distributions for the topics. We must determine them from a corpus of stories annotated with topics. We use a modified version of the EM method to determine the distributions. Given a new document, we could determine the most likely set of topics by considering all possible sets of topics exhaustively. Of course, this is too expensive. Instead, we first determine which individual topics are likely enough and then we consider all combinations of those few topics.

These topics can be used to provide a very concise summary of a story, or as a common set of key terms to be used in searching for documents about a desired combination of subjects without having to know what words are actually in the document.

Unsupervised Topic Discovery (UTD)

The cost of annotating a large corpus with thousands of distinct topics is fairly high. In addition, human annotators usually fail to indicate all of the relevant topics for each document. It would be desirable to determine the topics in any new domain or language automatically, given only a large corpus in that domain and language. We call this process Unsupervised Topic Discovery (UTD). The system must determine the set of topics in the corpus. For each topic, it must determine the distribution of words (the model of the topic) and it must also generate an understandable name (topic label) for the topic. The name must enable a person to guess what a topic refers to (without looking at the word distribution for the topic).

Topics as Basis Sets of Words.

We can define topics as statistical basis sets of words (or multi-word terms), such that any document can be constructed from a relatively small set of these basis sets. We

would like to find the basis sets that maximize the probability of the corpus under some reasonable constraint (e.g., you can't define each word its own basis set).

Document Clustering

We initially tried to use a document clustering method developed under the Topic Detection and Tracking (TDT) project [3]. The detection problem in TDT requires mutually exclusive clusters. We ran the TDT detection system multiple times with different thresholds to produce a tree of documents. We hoped that each node in the tree could be interpreted as a topic. But we found that the clusters created by this approach were not appropriate, since topics are, by their nature, not hierarchical. We need clusters that provide a many-to-many mapping from documents to topics.

Topics as Shared Themes

To develop an algorithm for finding topics we need to consider the nature of topics. Topics are themes that are shared among different documents, which in turn, may have different combinations of topics. (That is, they might be about different events that have some topic in common.) Any particular document will use only a small fraction of the words that could be associated with each of its topics. Using these principles, we developed the following initial algorithm for finding topics.

1. For each document in the corpus, find the "nearest" N other documents (say N=100).
2. For each pair of similar documents, find the intersection of terms in the two documents.
3. Cluster the document intersections into the desired number of topics.
4. Purify the topic distributions using EM.

Each of these steps is described in more detail below.

1. Find closest documents to each document

We treat each document as a query in order to find the most similar other documents, under the assumption that they must share at least one topic. We use the BBN Golden Retriever algorithm [2]. This probabilistic IR system creates a generative model from each document in the corpus. The model for the document is a two-state HMM for each document, where one state consists of the distribution of terms in the document, and the other is the General Language distribution. We find the N documents that were most likely to have produced the particular document, but also we discard all documents whose score is more than a few standard deviations below the maximum.

2. Document intersections.

We assume that if two documents share a topic, then they are also likely to share some words related to this topic. Some of the words in common are not important. This is dealt with in step 4, below. It is also likely that these two documents share *more* than one topic. We will discuss this problem later. With a corpus of 40,000 documents and 100 intersections for each, we have 4 million document intersections.

3. Cluster the document intersections.

This is the heart of the algorithm. There are many clustering algorithms. We use a k-means algorithm. First we choose a random set of intersections as initial (seed) clusters. We construct a model for each cluster from the words in the intersection, smoothed with a General Language state. Then we classify each of the intersections to the nearest (most likely) of these seed clusters. After the classification stage, we merge all of the intersections in each cluster into a single distribution. We iterate a few times. The result is a set of clusters of words, which we take to be topics.

4. Purify distributions EM.

Even though we remove stop words at the start, there are often a few thousand other common words that linger. We use the EM procedure of OnTopic™ to remove these common words, by modeling them with a General Language state. Each topic typically contains 100-200 terms.

Topic Name Assignment

At this point we have thousands of distributions of words. They can be used to perform topic spotting on documents in this corpus or another corpus in the same domain. But we also need to have mnemonic names for each topic cluster, or a person will not be able to know what they mean. How can we choose a meaningful name without true intelligence? For the moment, we have limited ourselves to choosing a name from among the terms in the topic distributions for each topic. But each topic distribution typically contains hundreds of terms.

We can learn how topic names are selected from existing corpora with annotated topics. We have used the Primary Source Media (PSM) corpus as an example. A one-year sample (July 1985 through June 1986) of this corpus contains 42,000 documents with about 5,000 distinct topics. Each document has an average of 4-5 topics, but the number ranges from 1 to 13. We used the OnTopic™ system to estimate topic distributions from this corpus. (Note that OnTopic™ does not use the name of a topic in any way.)

Then, we compare the terms in each topic distribution with the name assigned by human annotators. We find all the terms that share words with the assigned name. These terms are considered “correct”. The other terms are considered “incorrect”. We find that over 95% of the topics have a name that has a common word with at least one of the terms in the distribution. Next, we build a classifier to try to estimate the probability that any given term in a distribution will be a part of the name for that topic. The classifier uses 11 features of the term. Some of the more useful features were:

a. The probability of the term given the topic,
b. The percentage of documents with that topic that contain that term,
c. Whether that term is a name of a particular category (person, location, organization),
d. How many words there are in the term.

We choose a small number of terms – typically two or three – that have the highest probability for being part of the name. We find that this process is able to find at least

one term that matches (i.e., overlaps with) the human assigned name 42% of the time. However, when we look at the names and compare them subjectively with the human assigned name, we find about 69% of them to be good names.

Modifications

We noticed three problems with the initial algorithm described above. First, there were often several topics that were quite similar. Second, we found a few topics that seemed unfocussed, containing a little bit of several seemingly unrelated topics. These problems were clearly a failure of the clustering procedure. Third, some of the topics really consisted of more than one elementary topic. For example, since there were many stories about the Olympics in Atlanta, we end up with a single topic for Atlanta Olympics, rather than two independent topics for "Atlanta" and "Olympics", which could be used together when appropriate or separately in combination with other topics.

To alleviate the first problem, we tried to remove near duplicates before each round of k-means clustering. We use the TDT detection algorithm with a tight threshold and merge seeds or clusters that are very similar. In considering whether two clusters were sufficiently similar, we also took into account whether they received the same topic names. The second problem (unfocussed topics) was harder to solve and we are still working on detecting them.

The third problem (combination topics) required some careful thought about the nature of topics. If two different topics always appear together in a corpus, there is no mechanism for an unsupervised algorithm to split them apart. But if they also occur in other combinations, we hope to able to distinguish that they are not just different subsets of the same topic.

We reduced the problem by using a document clustering procedure at the very start of the procedure. Many of the close document pairs actually discuss the same event. Therefore, they share *several* topics. We added a preprocess in which we clustered the original documents. This grouped very similar documents (that probably discussed the same events). We then considered these clusters of documents as single documents in the original procedure. This enabled the algorithm to look for common topics among *different* events. This pre-clustering step also reduced the number of documents by a factor of two to three and the number of similar pairs for each event document by a factor of three. Thus, the number of document intersections produced was much smaller, which decreased the time needed for clustering them.

Results

It is not easy to evaluate the UTD process, since we do not expect the unsupervised process to match the human topics in the corpus exactly. Although it is not yet as good as human topic assignment, in many ways it is superior. In particular, the human annotators frequently omit some perfectly reasonable topics for a document. (They get tired after assigning a few topics). The computer version doesn't have this problem.

We tried using an objective measure (adapted from the TDT project) that compares the document assignments with those of the original PSM corpus. For each topic in PSM, we find the topic cluster that is most similar. Then, we measure the number of

documents missing from that cluster (documents that have that PSM topic label but are not in this cluster) and the number of extra documents (documents in this cluster that do not have that PSM topic). We average the percentage missing and extra documents across the true topics with equal weight. We found an unweighted average of 30% missing documents and 75% extra.. (These numbers would translate into approximately 70% recall and 50% precision.)

However, this result is not as bad as it seems. When we use the *supervised* training procedure on the PSM corpus, we found that the algorithm often assigned topics that were not assigned by the human annotators. For example, for the top-choice topic, the precision was approximately 75%. The precision of the 5^{th} choice topic was only about 40%. This means that about 60% of the topic assignments did not match any of those given by the human annotators. However, when we examined the top-choice topics subjectively, we found that, in most cases, the human annotator had omitted a perfectly reasonable topic and the system was correct. So we felt that we could not trust the human annotation in computing performance.

Subjective Evaluation

Although it is more expensive (and not repeatable), we considered using a subjective evaluation. We asked four subjects to compare the names of the UTD topics found for a story with the story itself or with the PSM topics. (Although the human annotators omitted topics, our human subjects could generally understand the nature of the story from the topics that were assigned.) We need to remember that part of the error is due to inappropriate topics and part of it is just due to inappropriate names. Each subject performed this subjective evaluation on 100 stories. We found that, for 66% of the stories, the top-choice topic assigned to the story was judged to be correct and aptly named. This result is very encouraging.

Needed work

This work is just beginning. There are many improvements to make. We do not believe a simple clustering algorithm will be sufficient to make all of them. For example, to remove clusters that contain different topics we will need to detect that they have parts that match better with other clusters. We believe there will be many applications for UTD in the future.

References

[1] R. Schwartz, T. Imai, F. Kubala, L. Nguyen, J. Makhoul, “A Maximum Likelihood Model for Topic Classification of Broadcast News”, Eurospeech-97, Rhodes, Greece, September, 1999.

[2] D. Miller, T. Leek, and R. Schwartz, ``A Hidden Markov Model Information Retrieval System," in Proceedings of the ACM Sigir '99.

[3] F.Walls, H.Jin, S.Sista, R.Schwartz. "Probabilistic Models for Topic Detection and Tracking", ICASSP'99, March 1999