CS365:Template based information extraction without using templates

Sachin Yadav
sachinky@cse.iitk.ac.in
Department of Computer Science and Engineering,
IIT Kanpur, India

February 26, 2013

1

Standard algorithms for template-based information extraction (IE) require predefined template schemas, and often labeled data, to learn to extract their slot fillers. This paper describes an approach to template-based IE that removes this requirement and performs extraction without knowing the template structure in advance. Our algorithm instead learns the template structure automatically from raw text, inducing template schemas as sets of linked events associated with semantic roles.

1.1

A template defines a specific type of event (e.g., a bombing) with a set of semantic roles (or slots) for the typical entities involved in such an event. Our goal in this paper is to perform the standard template filling task, but to first automatically induce the templates from an unlabeled corpus.

We learn templates by first clustering event words based on their proximity in a training corpus
Cluster the syntactic functions of these events based on selectional preferences and coreferring arguments.

An example snippet from a bombing document is given here:
The terrorists used explosives against the town hall. El Comercio reported that alleged Shining Path members also attacked public facilities in huarpacha, Ambo, tomayquichua, and kichki. Municipal official Sergio Horna was seriously wounded in an explosion in Ambo.

The entities from this document fill the following slots in a MUC-4 bombing template. Perp: Shining Path members Victim: Sergio Horna Target: public facilities Instrument: explosives

Learning Templates from Raw Text We approach this problem with a three step process: (1) cluster the domains event patterns to approximate the template topics, (2) build a new corpus specific to each cluster by retrieving documents from a larger unrelated corpus, (3) induce each templates slots using its new (larger) corpus of documents.

Information Retrieval for Templates Learning a domain often suffers from a lack of training data.
Our retrieval algorithm retrieves documents that score highly with a clusters tokens. The document score is defined by two common metrics: word match, and word coverage.
A documents match score is defined as the average number of times the words in cluster c appear in document d.
We define word coverage as the number of seen cluster words.

Inducing Semantic Roles Syntactic Relations as Roles
We want to cluster all subjects, objects, and prepositions.In the sentence, he ran and then he fell, the subjects of run and fall corefer, and so they likely belong to the same scenario-specific semantic role.We applied this idea to a new vector similarity framework. We represent a relation as a vector of all relations with which their arguments coreferred.
Clustering syntactic Functions
Cluster similarity is the average link score over all new links crossing two clusters.Clustering stops when the merged cluster scores drop below a threshold optimized to extraction performance on the training data.The first assumes that the subject and object of a verb carry different semantic roles. For instance, the subject of sell fills a different role (Seller) than the object (Good). The second assumption is that each semantic role has a high-level entity type. For instance, the subject of sell is a Person or Organization, and the object is a Physical Object.

Template Evaluation We now compare our learned templates to those hand-created by human annotators.

Information Extraction: Slot Filling We consider each learned semantic role as a potential slot, and we extract slot fillers using the syntactic functions that were previously learned. Thus, the learned syntactic patterns (e.g., the subject of release) serve the dual purpose of both inducing the template slots, and extracting appropriate slot fillers from text.

A document is labeled for a template if two different conditions are met: (1) it contains at least one trigger phrase, and (2) its average per-token conditional probability meets a strict threshold. Both conditions require a definition of the conditional probability of a template given a token.

Trigger phrases are thus template-specific patterns that are highly indicative of that template. After identifying triggers, we use the above definition to score a document with a template.
A document is labeled with a template if it contains at least one trigger, and its average word probability is greater than a parameter optimized on the training set. A document can be (and often is) labeled with multiple templates.

Entity Extraction Once documents are labeled with templates, we next extract entities into the template slots. Extraction occurs in the trigger sentences from the previous section. The extraction process is two-fold:
1. Extract all NPs that are arguments of patterns in the templates induced roles.
2. Extract NPs whose heads are observed frequently with one of the roles.