Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Joint Learning of Structural and Textual Features for Web Scale Event Extraction

Julia Wiedmann
Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, OX1 3QD Oxford
julia.wiedmann@cs.ox.ac.uk

Introduction

The web has become the central platform and marketplace for the organization, propagation of events and sale of tickets of any kind. Such events range from concerts, workshops, sport events, professional events to small local events. Individual's event choices vary tremendously based on preferences and lifestyle. Online users use the web to inform themselves about new events near their location of choice, and potentially use the website to purchase tickets or make a reservation for such events. Event Extraction from the web is a particularly difficult type of information extraction, dealing with the detection of specific types of events and its attributes mentioned in source language data. Traditional research in event extraction focuses on the extraction of political, cultural, or other general interest events from text. Such text is typically editorial news content, e.g (Kuzey, Vreeken, and Weikum 2014), or more lately from social media such as Twitter, e.g. (Ritter, Etzioni, and Clark 2012). This research, however, covers events presented in tables, lists, or most crucially on single pages devoted to that event. This thesis focuses on both the discovery and extraction of such "single event pages".

This work is inspired by a series of works on inducing wrappers for the extraction of specific document types from the web. For example, (Wang et al. 2009) proposes an approach for learning to extract news articles and their basic attributes from a very small training corpus. Though inspired by this work, the approach presented here differs considerably, in scope and techniques used. In scope, I am targeting events, which carry many more attributes than the document types targeted in the above work, and where attributes may occur both in the template structure (as in (Wang et al. 2009)) and in the event descriptions. Further, my approach balances the need for more training data due to the more complex domain with semi-supervised methods for acquiring that training data.

Problem Statement and Proposed Approach

The problem of event discovery and extraction is multifold: Events and their attributes are buried in the depth of event aggregators, spread across the web and thus hard to find. Each website may require a different access pattern to obtain the pages describing a single event ("single event page"). Within a given single event page, the event attributes have to be located, both within the structure of the page template and within the event description.

Such single event pages are typically split into a textual event description and a set of core *event attributes*, such as title, location, or time, that are highlighted and presented in the same template for all events of a particular source. In this work, I aim to learn a joint model for the extraction of event attributes from both event descriptions and templates.

The discovery of single event pages requires automated interaction with the website, such as form filling, to obtain all the relevant events hosted by the website. Different event aggregator websites are used by organisers to sell tickets, individual users to resell their tickets and in general to distribute information about events. Yet, there are not only big aggregators, such as Ticketweb.com, Eventbrite.com and Ticketmaster, but also a long tail of small event websites which host smaller and more local events. Thus,in addition to the extraction of event attributes, I also investigate the automatic discovery of event sources and single event pages within event sources. By considering all three problems as part of an integral system, I can exploit mutual reinforcement between the models derived for each sub problem.

In order to approach this problem, I have designed a framework for jointly learning models for extracting structural and textual event attributes from single event pages with minimal supervision. Unfortunately, labeled training data for this process is hard to come by and expensive to create. Therefore, the framework is complemented by a semi-supervised process for reducing the cost of acquiring labeled example pages. This process is a bootstrapping approach that uses seed values for each event attribute to annotate structural event attributes, eliminating noise by exploiting techniques from unsupervised template discovery (Crescenzi, Mecca, and Merialdo 2001).

Event Page Discovery

The event page discovery phase consists of finding relevant websites ("sources") for events and locating single event pages within those websites. Multiple approaches are combined in this step, to minimize supervision.

First, a seed set of events and event attribute instances is

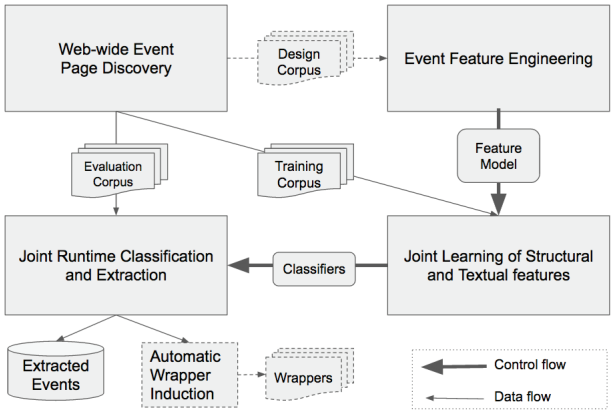

Figure 1: Overall Architecture

built up by combining relevant knowledge bases and creating crawlers for a small set of representative sources manually. These crawlers are created using OXPath (Furche et al. 2013), extending XPath with actions for form filling, iterations and markers for data extraction. In a second expansion step, this seed data set is further extended automatically by identifying single event pages in the Common Crawl, a repository of crawled web data, based on Microdata annotations and the annotations derived from the seed data. An initial run of the event discovery phase has yielded a data set of more than 800k single event pages from the OXPath Crawl, and around 2M pages from the Common Crawl.

Page Annotation

Following, the discovery of single event pages, two different techniques are used for the annotation phase of this project. Firstly, Microdata such as Schema.org is used to find relevant event attributes in the template structure of the page. These are annotations provided by the owner of the event source, including event specific attributes and types, such as title, location, start time and address of an event. They are further verified using the information from the seed data set. For cases where Microdata is not available, I use the seed data to annotate the website — through GATE's (Cunningham et al. 2002) NER framework. These annotations are verified and only accepted, if they pass stringed constraints, e.g., that an entire HTML node is annotated and that there are no conflicting annotations. Where they don't pass, I plan to use a limited amount of supervision to increase the training data set, if necessary. The annotation process incorporates many different Natural Language Processing (NLP) tools, such as NERs, heuristic rules and gazetteers, similar to the annotation layer of (Furche et al. 2012).

Since the Machine Learning model learns structural and textual features simultaneously, both types of features need to be annotated. In the case of Microdata, the the location on the website is given and therefore gives a structural indication of the node. In the case where Microdata is not available and the attributes on the single event page have been scraped with OXPath, the OXPath expression and the span within that node can be used to annotate the text.

Feature Engineering and Machine Learning

The above stated annotations form parts of the training and evaluation corpus and are used as an input for the training phase of the Machine Learning algorithm. This also means, that features for the annotated feature model have to be tweaked over time to fine tune the algorithm.

The exact feature model is still being revised, but initial testing shows potential in the combined use of textual and structural features. Furthermore, these tests suggested a combination of classification or structured prediction methods as a natural first step. The output of the Machine Learning phase is a classification model that supports three different types of classifiers: A page level classifier, which determines the template of a page, a node level classifier, which determine the location of the attribute within a given site, and a text level classifier, which determines which part of the node's text comprises the event information that we seek.

Runtime Classification and Extraction

The initial stage in the application of the trained Machine Learning model, is to cluster the unseen webpages per template. This permits the application of the node-level and textual classifiers in a site-wise iterative manner per template cluster. The output of the system are extracted events, with their attributes in an event database for further processing.

Conclusion

This work in event discovery and extraction from single entity pages contributes to the overall body of work in template-independent web data extraction. Future work aims to demonstrate the viability of this end-to-end approach, by applying it other other domains (e.g. product pages), as initial indicators suggest that the framework can be applied to other domains with relative ease.

References

Crescenzi, V.; Mecca, G.; and Merialdo, P. 2001. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*.

Cunningham, H.; Maynard, D.; Bontcheva, K.; and Tablan, V. 2002. A framework and graphical development environment for robust nlp tools and applications. In *ACL*.

Furche, T.; Gottlob, G.; Grasso, G.; et al. 2012. Diadem: Domain-centric, intelligent, automated data extraction methodology. In *WWW*.

Furche, T.; Gottlob, G.; Grasso, G.; Schallhart, C.; and Sellers, A. 2013. Oxpath: A language for scalable data extraction, automation, and crawling on the deep web. *The VLDBJ*.

Kuzey, E.; Vreeken, J.; and Weikum, G. 2014. A fresh look on knowledge bases: Distilling named events from news. In *CIKM'14*. ACM.

Ritter, A.; Etzioni, O.; and Clark, S. 2012. Open domain event extraction from twitter. *KDD '12*.

Wang, J.; Chen, C.; Wang, C.; Pei, J.; Bu, J.; Guan, Z.; and Zhang, W. V. 2009. Can we learn a template-independent wrapper for news article extraction from a single training site? In *KDD'09*.