Knowledge Base Population and Visualization Using an Ontology based on Semantic Roles

Maryam Siahbani, Ravikiran Vadlapudi, Max Whitney, and Anoop Sarkar
Simon Fraser University, School of Computing Science
8888 University Drive
Vancouver, BC. Canada.
{msiahban,rvadlapu,mwhitney,anoop}@sfu.ca

ABSTRACT

This paper extracts facts using "micro-reading" of text in constrast to approaches that extract common-sense knowledge using "macro-reading" methods. Our goal is to extract detailed facts about events from natural language using a predicate-centered view of events (who did what to whom, when and how). We exploit semantic role labels in order to create a novel predicate-centric ontology for entities in our knowledge base. This allows users to find *uncommon* facts easily. To this end, we tightly couple our knowledge base and ontology to an information visualization system that can be used to explore and navigate events extracted from a large natural language text collection. We use our methodology to create a web-based visual browser of history events in Wikipedia.

1. INTRODUCTION

Many machine reading approaches have been focused on dealing with the knowledge acquisition bottleneck. They read vast quantities of text and adopt a "macro-reading" approach [10] that attempts to find semantic relations between entities observed via a large number of instances of relations extracted from the shallow parsing of the text. This is true of systems with a rich ontology with sophisticated types such as NELL [5] or PROSPERA [12] or those with token-level facts such as TextRunner [1] which uses Freebase [3] to enable lightweight types when searching for information in such a knowledge base. The "macro-reading" approach extracts knowledge from large scale natural language data sets using shallow parses and frequent and predictive extraction patterns. However, the relatively shallow parse of the text and the noisy nature of bootstrapping means that a lot of information in each sentence is not recovered (even when these methods are augmented with a full dependency parsing of text [9]).

In this paper we focus on a "micro-reading" approach which uses a supervised semantic parsing approach to carefully extract all possible predicates from each sentence and extract all of its arguments. We use *Semantic Role Labeling* (SRL) [7] to extract predicate argument structures from text to automatically populate an ontology for a domain. Unlike NELL or TextRunner we use a fully supervised semantic parser trained on annotated data such as the Penn Treebank and PropBank. We use data sets that are smaller than the "macro-reading" approach but still large enough that human readers would not be able to conveniently read through the entire material. Thus, our system enables a *sense-making* process [14] for users that wish to navigate information encoded as natural language in a particular data set.

To validate our approach, we chose a specific domain, namely events that have occurred throughout human history as specified by articles in Wikipedia. This domain has facts about events, very few opinions, and sentiment is not an issue. The events ideally describe who did what to whom, when and where which is retrieved using the sentence predicate-argument structure.

Choosing Wikipedia means there is also a rich set of databases and knowledge bases associated with it such as DBPedia [2] and YAGO [17]. In the spirit of some previous work [19], we leverage such resources to augment our natural language semantic parser with rich semantic knowledge to get another view on the veracity of identifying of entities as different types such as locations and persons.

Our goal is to use a knowledge base constructed from Wikipedia text in order to visualize events in human history using three connected interactive visualizations: *faceted browsing* of all the entities and predicates we extract from the text and combine with Wikipedia information, a *timeline* of all events and a *map* that shows recovered locations of each event. We call this system Lensing Wikipedia (`http://lensingwikipedia.cs.sfu.ca`).

2. DATA PROCESSING

We used web pages from Wikipedia summarizing about 3500 years of human history. English Wikipedia contains about 2600 URLs (from 1500 BC to 2013) which are natural language summaries of important events in each year or decade in human history. We crawled all these Wikipedia URLs and obtained about 41,000 events, with each event described by one to several sentences. It provides 83,000 predicate argument structures determined by SRL approach.

The information extraction process is broken down into dif-

ferent steps of data extraction and alignment: using SRL approach to generate a *predicate argument structure* for each sentence (section 2.1), identifying entities using Name Entity Recognition (NER) and Wikipedia link structures (section 2.2) and then extracting temporal and spatial information for each event (section 2.3).

2.1 Semantic Roles and Predicate-centered Ontology

To extract predicate-argument structures, we use an SRL approach based on large-scale statistical machine learning [7, 8] based on a semantic role data set annotated by linguistic experts called the Proposition Bank corpus (PropBank) [13]. The following is an example of a PropBank style SRL annotation of a sentence.

Input: *In the opener, Sony Corp. would agree to buy Columbia Pictures Entertainment Inc. in a transaction valued at close to $ 5 billion.*
Semantic role labeling's output:
A0 (Buyer): *Sony Corp.*
Pred V (Buy): *buy*
A1 (Thing Bought): *Columbia Pictures Entertainment Inc.*

The SRL tool provides the predicate argument structure for sentences in the text such as *The House of York defeats the House of Lancaster* where *defeat* is the predicate with arguments *The House of York* (*arg0*) and *the House of Lancaster* (*arg1*). In our system, the semantic role labels (*arg0* and *arg1*) are converted into human readable types (*arg0:'entity victorious'* and *arg1:'entity defeated'*) using the information contained in the frames files in PropBank. Doing this automatically involves learning a mapping between abstract semantic role labels and verbose descriptions. This task is harder than it seems, because the verbose label depends on the sense of the verb. For instance, 'get' might have 'receiver' as verbose label for 'arg0', but it might also have 'instigator' for another sense of the verb (get across). We have worked on many different models to solve this task achieving an accuracy of 92%.

We noticed that these verbose labels provide a lightweight ontology or useful types that help search for specific events in the knowledge base we extract. For instance, we could search for all entities that match the "buyer" or "entity defeated" types. In particular, these predicate-centric types are much more useful than the types from Freebase [3] which need further disambiguation to be applied correctly to the entities and events we have in our knowledge base, especially when users browse events using all the Freebase types.

To automatically label entities with types, we create the training data as follows: Each predicate token in the PropBank is assigned a sense identifier that allows us to match the argument of that predicate to a detailed natural language description about that argument stored in the frames directory of PropBank for the predicate in question. The training data has 90,819 predicate instances and our dev (Sec. 24) and test (Sec. 23) sets have 3252 and 5273 instances respectively. Using this data, we undertake the task of *Verbose Label Prediction*, which (as far as we know) has not been a direct subject of a detailed experimental study before (although some SRL systems [15] pick a default verbose label in their web-based SRL tool). Some have also done verb-sense disambiguation on SRL output [20] which overlaps with our task but is not identical to it.

We compare against the following heuristic baseline methods:

Baseline-1: For an argument, say Arg0, assign the most frequent verbose label across the whole PropBank where frequency is defined as the number of occurrences in the PropBank as a whole. This baseline exploits the fact that verbose labels can remain same even if predicate sense varies.
Baseline-2: For an argument, say Arg0, assign the most frequent verbose label among all the verbose labels for that argument in the list of predicate frames. This baseline pays attention to the predicate when choosing the verbose label.
Baseline-3: Assign the first sense '01' for each predicate and return the verbose label for that argument in this frame. This technique is currently used in the UIUC SRL tool.
Baseline-4: A predicate frame in the PropBank is a list of arguments for a predicate. We take the list of arguments from the SRL output for each predicate and find the longest match for this list with the frame for each sense of this predicate. For each argument of the predicate in the SRL output, we return the verbose label found in this particular frame. We break ties by picking the predicate sense that has a lower integer identifier.

One way of solving the verbose label prediction problem is by reducing it to predicate sense prediction. The predicate sense prediction task maps to a multi-class classification task where given a set of senses for a predicate we pick one right sense which mainly depends on its context. The context information to predict a predicate sense can use features over the parse tree. We also use predicate level features like lemma, root form, voice and number of Senses and some contextual features like POS tags, chunk tags in a defined window.

In addition to lexical and contextual features, we extend an approach of transforming a sentence centered at a predicate to canonical form using hand-crafted transformation rules defined in [18] to predict its sense. A canonical form is a representation of a verb and its arguments that is abstracted away from the syntax of the input sentence. For example, "A car hit Bob" and "Bob was hit by a car" have the same canonical form, Verb = "hit", Deep Subject = "a car", Deep Object = "Bob". The rule transformation capture the structural information, such as position of arguments in the tree, presence/absence of arguments which are useful for predicting the sense.

A canonical form transformation rule consists of two parts: a tree pattern and a series of transformation operations. It takes a parse tree as input, and outputs a new transformed parse tree. The tree pattern determines whether the rule can be applied to a particular parse and also identifies what part of the parse should be transformed. The transformation operations actually modify the parse. Each operation specifies a simple modification of the parse tree. The algorithm for canonical form generation of a syntactic parse tree P is as follows: let S be a set of trees initialized to P, R be the set of rules. One iteration of the algorithm consists of

applying every possible matching rule $r \in R$ to every tree in S, and adding all resulting trees back to S. Rule matching is done top-down; find node that matches the constraints on the root of the tree pattern, then match the children of the root and then their children, etc. The rule set is carefully designed such that no new trees are added with repeated iterations. This simplification is done irrespective of verb hence this process needs to be done only once per sentence. Naive implementation of this algorithm would result in an exponential number of transformed parses and each such transformation iteration would require copying the whole parse. To alleviate these issues, we make use of an AND-OR tree for storing all transformed trees (S) as defined in [18].

We evaluate performance of our approaches at two levels, predicate sense prediction and verbose label prediction. Evaluation measure for predicate sense prediction task is simply the total number of times a correct sense is predicted by total number of predicates and for verbose label prediction, how often does an identified argument gets a correct verbose label. This type of evaluation is chosen to evaluate the performance of verbose label prediction irrespective of the SRL tool performance. Table. 1 summarizes the performance of our approaches. The model which uses canonical form transformation rules (*Transform*) as an additional feature performed marginally better than the standard features (*Standard*) and both significantly outperform the heuristic baselines.

Approach	**Section-24**	**Section-23**
Predicate Sense Prediction		
UIUC Baseline	82.8	82.3
Standard	90.3	90.1
Transform	90.5	90.3
Verbose Label Prediction on UIUC System Output		
Baseline-1	11.7	11.9
Baseline-2	60.9	57.8
UIUC Baseline-3	91.46	90.51
Baseline-4	93.4	92.6
Standard	**94.7**	**93.9**
Transform	**94.85**	**94**

Table 1: Predicate Sense Prediction using PST on Sec. 23 & Sec. 24 of PropBank

predicate	**Freq**	**Arg0**	**Arg1**
kill	2100	killer	corpse
found	1801	agent	thing set
defeat	1637	entity victorious	entity defeated
succeed	1350	entity succeeding	task
lead	1032	leader	thing led

Table 2: Most frequent predicates in human history WikiPedia articles.

Each sentence might have multiple predicates, each with multiple arguments. We use only the first two arguments (*arg0* and *arg1*) for each predicate. Table 2 shows the 5 most frequent predicates in the data along with their frequencies, arg0 and arg1.

2.2 Entity Extraction

The knowledge acquisition consists in name entity extraction from text data. We have done this in two ways: using Wikipedia hyperlinks and name entity recognition (NER). LensingWikipedia focuses on *Person* and *Location* as entity types. We used Stanford NER [6] to detect candidate entities along with their types. Each candidate entity is verified by mapping to an article in Wikipedia.

In addition to entities recognized by NER we take the advantage of hyperlinks embedded in the event's texts. Wikipedia is a wide coverage resource of notable entities. Each article is uniquely identified by the most notable name of the subject described in that article[1]. Entities with existing article page are supposed to be linked whenever they are mentioned in Wikipedia articles (at least the first mention of the related entity in each article). It provides a rich resource to extract related entities without disambiguation.

Using NER and Wikipedia hyperlinks, we come up with a set of candidate entities associated with their Wikipedia articles but still need to be identified as correct entity type (person, geographic location or NIL). Each Wikipedia article is referenced under some categories, often fine-grained and specific. Wikipedia categories are organized as a hierarchical ontology but they are not anchored in general conceptual classes like entity types (e.g. person). Some heuristics based on categories and infoboxes are used to verify and map articles to entity types (e.g. categories like *Category:y_births* where y is a year, are manually associated with entity type *person*). By combining NER output with Wikipedia structured information we obtain $\approx 12K$ persons and $\approx 12K$ locations.

2.3 Temporal and Geographical Identification

From the semantic parse and the URLs we extract information such as the date when the event occurred. For event geo-location, we used the Wikipedia articles associated with each location entities extracted in previous step to obtain the latitude and longitude information. Having geo-location information, we are able to extract the current country(ies) where events happened by reverse geocoding. We used google geocoding api for reverse geocoding. An example of the final representation is in Fig. 2. Each event could contribute several such entries in our transformed data-set.

```
"arg0":      "Emperor Le Thanh Tong",
"arg1":      "the Champa Capital",
"event":     "capture",
"latitude":  21.03,
"longitude": 105.85,
"country":   "Vietnam",
"roleArg0":  "getter",
"roleArg1":  "thing gotten",
"year":      1471,
"person":    "Le Thanh Tong"
```

Figure 2: Output of NLP plus temporal identification and geolocation for the event description: "March 1 - Emperor Le Thanh Tong captures the Champa Capital, establishing new regions in middle Vietnam."

3. VISUALIZATION

To show the effectiveness of this ontology to represent the underlying data, we created an interactive visualization interface using obtained ontologies called "Lensing Wikipedia". We leverage three connected visualizations components: geographical view (map), a temporal view (timeline) and a

[1]For ambiguous names, additional information placed in parentheses, e.g. Michael Jordan, Michael Jordan (footballer).

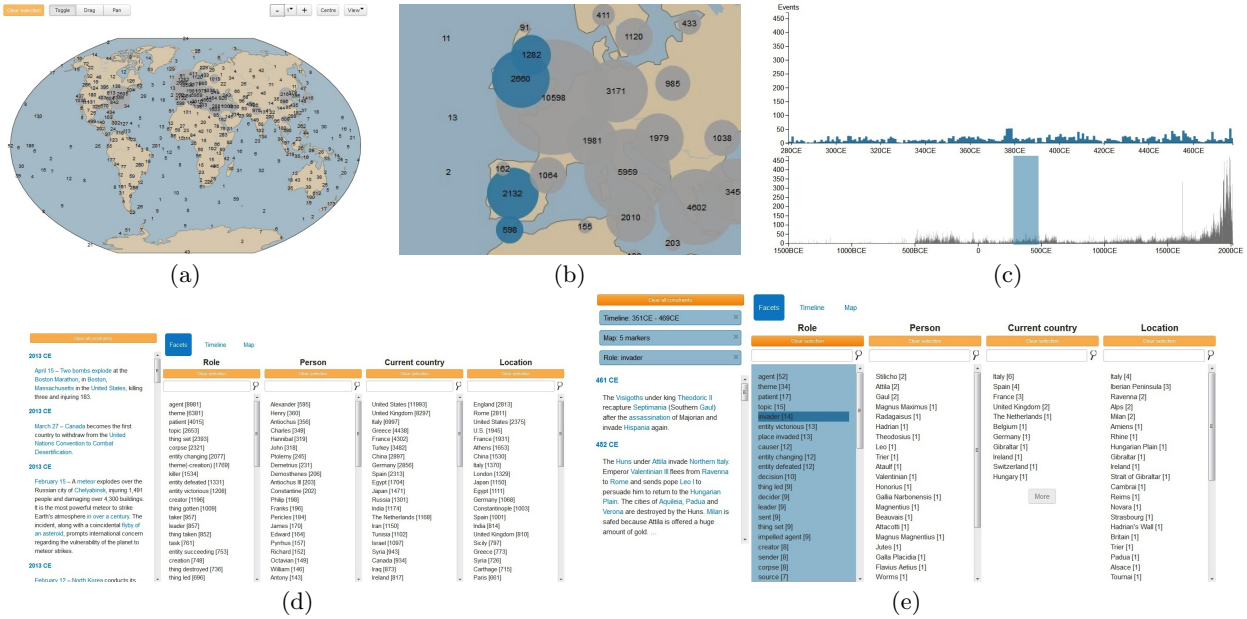

Figure 1: Visualizations of events in faceted browsing, time and space. (a): Map view (flat). (b): Applying spatial constraint by selecting clusters in the map view. (c) Global timeline (downside) and local selection of time interval. (d) Lensing Wikipedia (e) constraints on time, place and role

faceted view using extracted entities and their roles in the history events.

Geo-location data allowed situating events on an interactive map (for which we used Natural Earth [11] and d3 [4]) shown in Fig. 1(a). Map has different views: flat, globe and butterfly. Clusters indicate number of events in a region depicted by size. User can easily move in the map, zoom in and select one or more clusters in the region of interest (Fig. 1(b)). Selecting clusters would load only region specific events.

The temporal identification of each event allowed us to use an interactive timeline. It shows global timeline and local selection of time interval simultaneously shown in Fig. 1(c) (we should emphasize that we are using information extracted from semi-structured text on Wikipedia for this).

Faceted view/browsing is a technique for accessing information organized according to a faceted classification system, allowing users to explore a collection of information by applying multiple filters. It has been applied for closed domain datasets like Nobel prize winners, and recipes [16]. To our knowledge, this is the first time faceted browsing has been implemented for an open domain dataset like history articles. We employ named entities and their roles (identified using predicate-argument structures) as facets which defines a unique classification of event descriptions. Entities (person, location, contemporary country of events) and theirs roles (identified by SRL) are used as facets to browse events (Fig. 1(d)). Each list is a facet. Choices in the list are added as a constraint which can be removed in any order.

The three views: map, timeline and facets are all interlinked. Facets interact with the map and timeline by showing data specific to a selected region and selected time range (Fig. 1(e)). Each element in these facets is a constraint and when clicked a constraint is added and all the events satisfying this constraint are displayed chronologically. In Fig. 1(e), the original event descriptions from Wikipedia are shown in the left-side bar. Descriptions are wikified by adding hyperlinks to the corresponding Wikipedia article.

Faceted browsing provides a flexible way of browsing data. Constraints can be quickly searched, added or removed in any order. With Named entities, predicates and roles as facets one can search using intuitive terms like ”attacker” or ”Alexander” and quickly narrow down on to a few related events with fewer constraints which otherwise is a tedious task. The faceted interface simultaneously shows where to go next and how to return to previous states and also provides free text search within the category structure at the same time. It provides an organizing context for results and for subsequent queries which is important for exploration and discovery.

4. OBSERVATION AND CONCLUSION

Some advantages of LensingWikipedia are: a) Focusing on a location with single click reveals a summary of its history from Wikipedia. b) It is potentially useful for Wikipedia editors to monitor Wikipedia coverage and add missing important events. c) Easy exploration of events, e.g., to find out more about a specific country, would require selecting the country in map or facets (”location” or ”current country”); and then by selecting different roles like “attacker”, “entity victorious” we can reach a few specific information about the country of interest. Furthermore the map view reveals all countries engaged in ’wars’ at a certain time. Additional information about the distribution of such events across time is provided by the timeline indicating active and passive time frames. d) To list out all ’invaders’ of a specific location requires just two clicks, selecting a location and the ’invader’ role on faceted view. We observed that our browser for Wikipedia provided valuable insights which could be easily obtained with a few clicks.

5. REFERENCES

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, Sept. 2009.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[4] M. Bostock. Data-driven documents. `http://d3js.org/`.

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, 2010.

[6] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[7] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288, Sept. 2002.

[8] Y. Liu and A. Sarkar. Experimental evaluation of LTAG-based features for semantic role labeling. In *Proceedings of the (EMNLP-CoNLL)*, pages 590–599, jun 2007.

[9] Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[10] T. M. Mitchell, J. Betteridge, A. Carlson, E. R. H. Jr., and R. C. Wang. Populating the semantic web by macro-reading internet text. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, 2009.

[11] NACIS members and cartographers around the globe. Natural earth. `http://www.naturalearthdata.com/`.

[12] N. Nakashole, M. Theobald, and G. Weikum. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 227–236, New York, NY, USA, 2011. ACM.

[13] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, Mar. 2005.

[14] P. Pirolli and S. Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of 2005 International Conference on Intelligence Analysis*, pages 2–4, May 2005.

[15] V. Punyakanok, D. Roth, and W. Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2), 2008.

[16] E. Stoica and M. A. Hearst. Automating creation of hierarchical faceted metadata structures. In *In Procs. of the Human Language Technology Conference (NAACL HLT)*, 2007.

[17] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA, 2007. ACM Press.

[18] D. Vickrey and D. Koller. Sentence simplification for semantic role labeling. In *ACL*, pages 344–352, 2008.

[19] F. Wu and D. S. Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[20] P. Ye and T. Baldwin. Verb sense disambiguation using selectional preferences extracted with a state-of-the-art semantic role labeler. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 139–148, Sydney, Australia, November 2006.