Cluster-based Web Summarization

Yves Petinot and **Kathleen McKeown** and **Kapil Thadani**
Department of Computer Science
Columbia University
503 Computer Science Building
1214 Amsterdam Avenue
New York, New York, 10027
{ypetinot|kathy|kapil}@cs.columbia.edu

Abstract

We propose a novel approach to abstractive Web summarization based on the observation that summaries for similar URLs tend to be similar in both content and structure. We leverage existing URL clusters and construct per-cluster word graphs that combine known summaries while abstracting out URL-specific attributes. The resulting topology, conditioned on URL features, allows us to cast the summarization problem as a structured learning task using a lowest cost path search as the decoding step. Early experimental results on a large number of URL clusters show that this approach is able to outperform previously proposed Web summarizers.

1 Introduction

Abstract Web summaries, which describe the topics and functionalities of Web pages at an *abstract* level, play an essential part in the discovery of new sites and services on the Web. Such summaries are intrinsically difficult to obtain using the content of Web pages. As such, the most successful methods for generating them are effectively extractive and based on the identification of likely abstractive content from linking pages. These *URL-centric* techniques however require a significant amount of redundancy in linking content (Delort et al., 2003).

In this paper, we propose a *summary-centric* approach to Web summarization based on the observation that summaries for similar URLs exhibit both similar content and structure. This similarity is apparent when analyzing summaries from a single ODP category, examples of which are shown in Table 1. We can see there that summaries of semantically related URLs tend to share common concepts and differ mostly at the level of target-specific attributes. Given a previously unseen URL U, such a cluster could thus be used as a source of potentially relevant terms for that URL's summary. In particular, these relevant terms include abstract terms, which may not otherwise be observed in the input data. We propose a graph-based summarization framework that can leverage this phenomenon.

2 Proposed Framework

Given a reference cluster C, we represent the space for summary generation as a graph $G_C = (V_C, E_C)$. This graph is obtained by fusing training summaries in C into a word graph. Each summary g_i is mapped to a path between the shared source and sink nodes. Each word g_i^j is thus mapped to a node N_k and each pair of neighboring words (g_i^j, g_i^{j+1}) to a directed edge (N_k, N_l). Notably, we add nodes as needed to guarantee that individual summaries are cycle-free. Figure 1 shows a simple summary graph.

2.1 Node Alignment

During the graph construction, nodes from distinct paths are iteratively combined to elicit the structural and content commonalities between summaries in the reference cluster. Following (Filippova, 2010) unambiguous nodes — i.e. nodes whose surface form match exactly and for which only one candidate exists — are always aligned, while ambiguous nodes are aligned to the candidate node with maximum context overlap. When there is no context overlap, a new node is added to the graph.

URL	Abstract Summary
http://www.qgazette.com/	Published weekly for the Queens, New York community. Includes information on politics, religion, dining, seniors, events, archives, classifieds and subscription details.
http://www.queenschronicle.com/	Local Queens NY news classifieds.
http://www.rockawave.com/	Published weekly in Rockaway, featuring local news, sports, community calendar, classified ad section, archives and subscription and advertising details.
http://www.observer.com/	Online version of the newspaper, providing coverage of local politics and media, Real Estate, fashion, and the Arts.
http://www.nytimes.com/	Online edition of the newspaper's news and commentary. [Registration required]

Table 1: Sample entries form the ODP category /News/Newspapers/Regional/United_States/New_York. All entries in this category describe sites of news organization located in the New-York area.

2.2 Template Slots

The content of reference summaries is likely to be only partially relevant to a previously unseen URL. In particular, certain paths in the summary graph may contain nodes whose surface form is target-specific. We use the following heuristic to decide on the presence of template slots in a summary:

- Adverbs, adjectives and named entities are treated as slots.
- Any term occurring in at least 25% of paths will not be treated as a slot;

Similarly to (Barzilay and Lee, 2003), slot identification is performed prior to alignment.

3 Features

In this section we present the feature sets used to condition the summary graph topology on a target URL U. Two aspects of the graph need to be trained, namely edge costs and slot locations. We introduce features for both.

3.1 Edge Feature Templates

We use the following feature templates to represent the compatibility between U — represented by its text modalities, as listed in Table 2 — and a specific edge in the summary graph.

Edge prior Probability of appearance of edge e_{ij} in reference paths (summaries).

Edge appearance + Modality Frequency of edge e_{ij} in each modality M_k. We consider that e_{ij} appears in M_k if its source and sink co-occur in M_k.

Source/Sink prior Probability of Source/Sink node N_i in reference paths (summaries).

Source/Sink appearance + Modality Indicates whether Source/Sink node N_i occurs in M_k.

Modality + n-gram Compatibility between the edge e_{ij}, the modality M_k and the n-gram n_l, where n is in the range $[1, 3]$.

3.2 Slot Features

To allow for the use of supervised methods in learning optimal edge costs we need a graph whose structure remains unchanged from one training instance to the next. The fillers of slot locations are thus described using features that are not surface-based:

Semantic Type We only consider fillers compatible with the host slot.

Modality appearance Frequency of filler candidate in modality M_k.

Content HTML Context HTML (Style + Structural) context of filler candidate in the content of U.

4 Learning Model

Using the summary graph topology and the features defined above we express the abstract Web summarization task as a structured learning problem. Specifically, we seek to obtain edge weights such that the cost of individual reference summaries — which, as we saw earlier, are mapped to paths — is minimized. Given a set of features $\mathcal{F}$, the optimal set of feature weights w^* is such that:

$$w^* = \arg\min_w \sum_{g \in \mathcal{R}(U)} Cost_w(g) \qquad (1)$$

Since our core constraint in building the summary graph is that each summary should map to a cycle-free path, identifying the optimal summary given a set of edge weights reduces to solving a lowest cost path problem:

$$w^* = \arg\min_w \sum_{g \in \mathcal{R}(U)} \sum_{i=1}^{|g|} \sum_{f_k \in \mathcal{F}} Cost_{f_k}(e_{g_{i-1}g_i}) \qquad (2)$$

Modality	Feature Types	Description
URL content	n-gram ($n \in [1,3]$) n-gram + context	n-grams generated from the target page content n-gram with HTML context (immediately surrounding HTML tag)
URL title	1-gram	1-grams generated from the target URL title
URL words	1-gram	1-grams generated from the target URL string (e.g. "nytimes", "com")
URL anchortext	n-gram ($n \in [1,3]$)	n-grams generated from the anchortext for the target URL

Table 2: Modality and features used to represent a target URL U.

In this setting, generation is achieved by running the decoder using the optimal set of edge weights.

4.1 Structured Perceptron

One way to solve this learning problem is to use a structured perceptron algorithm (Collins, 2002) where weights w control the linear contribution of individual features to the aggregate cost of edges.

$$Cost_{f_k}(e_{ij}) = w_k^{e_{ij}} \cdot f_k \quad (3)$$

The structured perceptron algorithm is shown in Algorithm 1. At every iteration, decoding is achieved via a search for the shortest path based on the edge costs induced by the current weights w^*. Following recent work on structured learning (Huang et al., 2012), we do not require this search to be exact, but to guarantee that each iteration results in a valid update. Our decoder thus uses beam search (beam size $b = 5$) combined with an early update procedure, the latter helping in significantly speeding up model training.

Algorithm 1 Structured Perceptron

Require: $\{u^i, g^i\}_{i=1}^n$
$w^* \leftarrow \{0\}$
for $i = 1 \rightarrow T$ **do**
  for $j = 1 \rightarrow n$ **do**
    $g^* \leftarrow ShortestPath_{Gw^*}(u^i)$
    $w^* \leftarrow w^* + \phi(u^i, g^i) - \phi(u^i, g^*)$
  end for
end for

4.2 Slot Filling

During the inference phase, we substitute slot locations with alternate paths, each containing a candidate filler for the slot. Each of these paths is associated with the slot features discussed earlier, however the feature weights of each slot are shared, thus allowing the learning algorithm to converge towards appropriate weights for filler selection.

5 Evaluation

We compare the performance of our model against a reimplementation of the two summarization algorithms - content-based and context-based - proposed in Delort et al. (2003). We apply our summarization algorithms and the baselines to a random sample of 56 ODP categories comprising at least 50 entries. For each category, we split the set of available summaries into training (90%) and testing (10%), train the summarization algorithms on the training set, and report (macro) average performance on the testing set.

ROUGE (Lin, 2004) results comparing both our proposed summarization model and the baseline systems to the ODP ground truth are provided in Table 3. The summary-graph model, both with and without slot-filling, shows significant improvements compared to the baselines in terms of ROUGE-1 and ROUGE-L scores. ROUGE-2 performance, however, is not on par with the baselines. Long-distance sequence similarity (ROUGE-L) being higher, we believe this could indicate the inability of our model to capture target-specific bi-grams that have little or no support in the training summaries. Allowing the topology of the summary-graph to adapt to its target, for instance by introducing missing edges supported by the input data, should help alleviate this issue. Finally, the performance of the model with slot filling shows slight improvement over the basic model, however we observed that the system frequently fails in extracting slot fillers. Future work will focus on acquiring more filler candidates and better features to model them.

6 Previous and Related Work

The work presented in this paper is linked to previous research in Web summarization and T2T language generation. Most works on the former have been on extractive methods, owing to the complexity of Web content but also to the need for compressed versions of Web pages. Other works have in fact eluded the question of generation to instead focus on the extraction of salient keywords from Web sites (Glover et al., 2002; Zhang et al., 2004). In the context of Web search engines, the compression task is constrained further by the amount of *SERP real estate* available for any single snip-

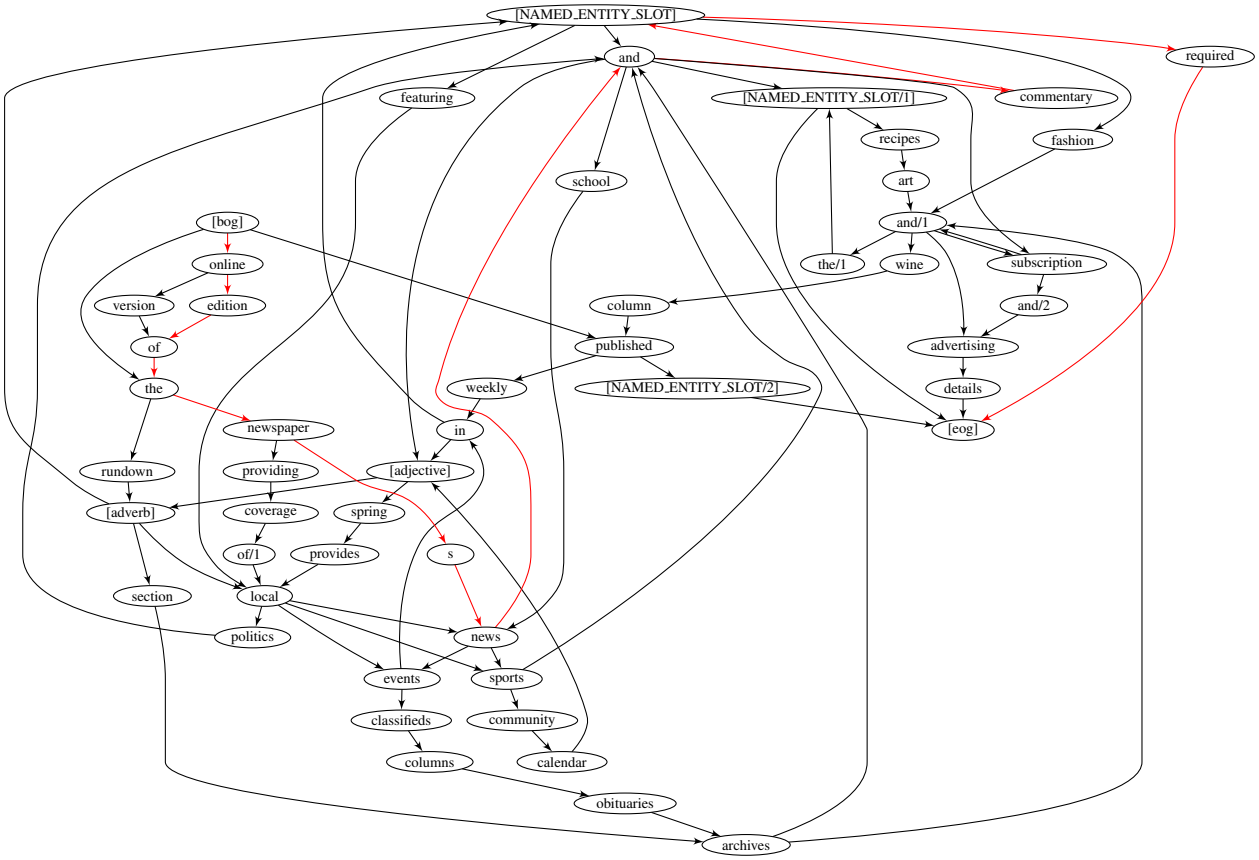

Figure 1: Summary graph associated with a subset of /News/Newspapers/Regional/United_States/New_York. The path for the URL `http://www.nytimes.com` is shown in red

Summarization System	ROUGE-1	ROUGE-2	ROUGE-L
Delort - Content	0.07163	0.04492	0.06574
Delort - Context	0.06783	0.03979	0.06197
Summary-graph	0.16222†	0.02775	0.14370†
Summary-graph + slot filling	0.16832†	0.02702	0.14729†

Table 3: Performance of summarization algorithms. † indicates statistically significant improvements (according to a paired t-test with $p < 0.05$) compared to the provided baselines.

pet and how content may be truncated (Clarke et al., 2007). Sun et al. (2005), in particular, leveraged the ODP hierarchy to mitigate data scarcity for certain URLs, however they did not exploit ODP summaries themselves. Several efforts have focused on producing Web summaries using the content of linking pages as a source of descriptive content. Amitay and Paris (2000) assumes full summaries can be readily found on a single page linking to the target site. Delort et al. (2003) makes less stringent assumptions and seeks to combine descriptive content from multiple linking pages. Closer to our work, Berger and Mittal (2000) proposed a generative solution embracing the noisiness of Web data and trained directly over ODP (URL,summary) pairs. Finally our work relates to T2T generation as we seek to generate well-formed sentences without resorting to semantic representations of either the input or output contents. Graph-based models similar to ours have been used for tasks ranging from string reconstruction (Wan et al., 2009) to sentence fusion and compression (Filippova and Strube, 2008; Filippova, 2010).

7 Conclusion and Future Work

We have introduced a word graph model for the task of Web summarization. and showed that per-cluster word graphs make it possible to combine abstractive and extractive behaviors. A limitation of our model is the need for existing reference clusters from which we build our summary graphs. Future work will investigate the dynamic production of such clusters.

References

Einat Amitay and Cecile Paris. 2000. Automatically summarising web sites - is there a way around it? In *CIKM 2000*, pages 173–179.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT, 2003*, pages 16–23.

A. Berger and V. Mittal. 2000. Ocelot: a system for summarizing web pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 144–151.

Charles L.A. Clarke, Eugene Agichtein, Susan Dumais, and Ryen W. White. 2007. The influence of caption features on clickthrough patterns in web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieva*, pages 135–142.

Michael Collins. 2002. Discrimative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, page ...

Jean-Yves Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. 2003. Enhanced web document summarization using hyperlinks. In *Hypertext 2003*, pages 208–215.

K. Filippova and M. Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, Hawaii, Association for Computational Linguistics*, page 177–185.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of COLING 2010*, pages 322–330.

Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. 2002. Using web structure for classifying and describing web pages. In *Proceedings of WWW 2002*, pages 562–569.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter for the Association for Computational Linguistics: Human Language Technologies*, pages 142–151.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26*.

Jian-Tao Sun, Dou Shen, Hua-Jun Zeng, Qiang Yang, Yuchang Lu, and Zheng Chen. 2005. Web-page summarization using clickthrough data. In *SIGIR 2005*, pages 194–201.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2009. Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 852–860.

Y. Zhang, N. Zincir-Heywood, and E. Milios. 2004. World wide web site summarization. In *Web Intelligence and Agent Systems, 2(1)*, pages 39–53.