Using Biased Random Walks for Focused Summarization

Güneş Erkan
Department of EECS
University of Michigan
Ann Arbor, MI 48109-2121
gerkan@umich.edu

Abstract

We introduce a graph-based sentence ranking algorithm for extractive summarization. Our method is a version of the LexRank algorithm we introduced in DUC 2004 extended to the focused summarization task of DUC 2006. As in LexRank, we represent the set of sentences in a document cluster as a graph, where nodes are sentences and links between the nodes are induced by a similarity relation between the sentences. Then we rank the sentences according to a random walk model defined in terms of both the inter-sentence similarities and the similarities of the sentences to the topic description.

1 Introduction

People often prefer to see some specific information about a topic in a summary rather than a generic summary that tries to cover as much of the information from the original documents as possible. An example summarization problem from Document Understanding Conferences (DUC) 2006 is as follows:

- topic: *international adoption*
- focus: *What are the laws, problems, and issues surrounding international adoption by American families?*

Given a set of documents about a topic (e.g. “international adoption”), the systems are required to produce a summary that *focuses* on the given aspects of that topic. Our approach to this problem is based on the LexRank framework (Erkan and Radev, 2004a). LexRank was originally proposed for the generic summarization problem and ranked one of the top systems in DUC 2004 (Erkan and Radev, 2004b). In this paper, we describe a *topic-sensitive* extension of LexRank that can handle topic descriptions in order to produce summaries that focus on a particular aspect of a topic.

2 LexRank: Graph-based Centrality as Sentence Salience

To compute LexRank, we first segment the documents into sentences, and then construct a graph where each node represents a sentence. The edge relation between the nodes is induced by a similarity metric. In a generalized form the LexRank equation can be written as

$$\mathrm{LR}(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{w(v,u)}{\sum_{z \in adj[v]} w(v,z)} \mathrm{LR}(v) \quad (1)$$

where $\mathrm{LR}(u)$ is the LexRank value of sentence u, and $w(v,u)$ is the weight of the link from sentence v to sentence u. We used the cosine measure for the edge weights $w(v, u)$ in DUC 2004. The LexRank Equation 1 is defined in a recursive manner, and can be computed via an iterative routine called the *power method*. An extractive summarization method that is almost equivalent to LexRank with cosine links was independently proposed by Mihalcea and Tarau (2004).

An interesting interpretation of the LexRank value of a sentence can be understood in terms of the concept of a random walk. A random walk on a graph is the process of *visiting* the nodes of the graph according to a specified *transition probability* distribution. Suppose we have a sentence similarity graph as described above, We define a random walk on this graph in such a way that it starts at a random sentence and then at each step, with probability d it jumps to a random sentence with uniform probability, with probability $1 - d$ it visits a sentence that is adjacent to the current sentence with probability in proportion to the outgoing edge weights of the current sentence. The LexRank value of a sentence gives us the limiting probability that such a random walk will visit that sentence *in the long run*. Equivalently, the LexRank value is the *fraction* of the time such a random walk spends on the particular sentence. The motivating assumption behind the LexRank method is that the information that is repeated many times in a cluster is the salient information that needs to be represented in a summary. Furthermore, if a sentence is similar to a lot of other sentences in a cluster, then it contains common information with other sentences that is also repeated in it; therefore it is a good candidate to be included in an extractive summary. Note that such a sentence will be strongly connected to a lot of other sentences in the similarity graph. The random walk we described above is more likely to visit a sentence that is better connected to the rest of the graph with strong links. Thus the LexRank value of such a graph will be higher.

3 Biased LexRank

There is nothing in Equation 1 that favors certain sentences based on a topic focus; LexRank is completely *unsupervised* in the sense that it only depends on the overall structure of the graph. The first term, $\frac{d}{N}$, is introduced to make the matrix ergodic so that a solution to the equation exists. It does not have a big impact on the final ranking of the nodes since it favors all the nodes equally during the random walk. With probability d, the random walk jumps to any node with uniform probability. This suggests an alternative view of the random walk process. We can combine more than one random walk models into one random walk process. Indeed, we could use a non-uniform distribution in combination with the random walk based on the weight/similarity function $w(\cdot, \cdot)$.

Suppose we have a prior belief about the ranking of the nodes in the graph. This belief might be derived from a baseline ranking method which we trust to a certain extent. For example, in the focused summarization task, we can rank the sentences by looking at their similarity to the topic description. Let $b(u)$ be the score of u based on this baseline method. We can *bias* the random walk based on $b(\cdot)$ while computing LexRank as follows:

$$\begin{aligned} LR(u) = {} & d \cdot \frac{b(u)}{\sum_{z \in S} b(u)} \\ & + (1 - d) \sum_{v \in adj[u]} \frac{w(v, u)}{\sum_{z \in adj[v]} w(v, z)} LR(v) \end{aligned} \tag{2}$$

where S is the set of all nodes in the graph. We call Equation 2 *biased* or *topic-sensitive* LexRank since it favors certain set of sentences during the random walk based on a prior distribution. When $d = 1$, $p(\cdot)$ ranks the nodes exactly the same as $b(\cdot)$. When $d < 1$, we have a mixture of the baseline scores and the LexRank scores derived from the *unbiased* structure of the graph. In other words, biased LexRank ranks the sentences by looking at the baseline method and the inter-sentence similarities at the same time. A version of biased LexRank was successfully applied to the sentence retrieval for question answering task (Otterbacher et al., 2005).

4 Using Generation Probabilities as Link Weights

As mentioned in Section 2, we used the cosine measure for the edge weights of the sentence similarity graphs in DUC 2004. Kurland and Lee (2005) proposed a document retrieval method that is similar to LexRank. The main differences of their approach from our original formulation is that they use documents instead of sentences, and they define the edge weight $w(u, v)$ from sentence u to sentence v as the *generation probability* of u given v. In this section, we explain the language model-based edge weights that we employed in DUC 2006 motivated by Kurland and Lee's work.

Given a sentence v, we can compute a (unigram) language model from it. A straightforward way of computing this language model is the maximum likelihood estimation (MLE) of the probabilities of the words to occur in v:

$$p_{ML}(w|v) = \frac{c_v(w)}{|v|} \tag{3}$$

where $c_v(w)$ is the number of times the word w occurs in v, and $|v|$ is the total number of words in v. The MLE is often not a good approximation for a language model since the words that do not occur in the text that we compute the word frequencies from get zero probability. This is even a bigger problem when we compute language models from relatively a short input text such as a sentence composed of few words. To account for the unseen words, we smooth the language model computed from a sentence using the language model computed from the entire cluster:

$$p(w|v) = (1-\lambda)p_{ML}(w|v) + \lambda p_{ML}(w|C) \tag{4}$$

where C is the entire document cluster. Equation 4 is often called *Jelinek-Mercer smoothing*. λ is a trade-off parameter between the MLE computed from the sentence and the MLE computed from the entire cluster. $p(w|v)$ is nonzero for all words that occur in the cluster provided that $\lambda > 0$.

We can also talk about the generation probability of a sentence given the language model computed from another sentence. For example,

$$p(u|v) = \prod_{w \in u} p(w|v) \tag{5}$$

defines the generation probability of sentence u given the language model of sentence v. Since the probabilities of all words get multiplied with each other, longer sentences tend to get smaller generation probabilities. Therefore, we normalize the generation probability of a sentence by its length:

$$\text{gen}(u|v) = p(u|v)^{\frac{1}{|u|}} \tag{6}$$

We use gen$(u|v)$ as the weight of the link from u to v in the graph-based representation of the cluster. Note that gen$(u|v)$ is not necessarily equal to gen$(v|u)$. The probability of a 1-step random walk (i.e. a random walk of length 1) from u to v is proportional to the (normalized) generation probability of u given the language model computed from v. If a sentence has strong incoming generation links in the graph, it is an evidence that the language model of that sentence can generate other sentences more successfully. Revisiting the random walk model of LexRank, the LexRank value of of a sentence is a measure of its *generation power*, that is, how likely it is to generate the rest of the cluster from the language model of the specific sentence in the long run.

Extending the use of generation probabilities to biased LexRank for the focused summarization task is straigtforward. For the baseline ranking method, we use the generation probability of the topic description from the sentences. A sentence is ranked higher if its language model can generate the topic description with a larger probability. Given a topic description t, the final score for a sentence u is computed by the following biased LexRank equation:

$$LR(u|t) = d \cdot \frac{\text{gen}(t|u)}{\sum_{z \in C} \text{gen}(t|z)} + (1-d) \sum_{v \in adj[u]} \frac{\text{gen}(v|u)}{\sum_{z \in adj[v]} \text{gen}(v|z)} LR(v|t)$$

5 Experiments and Results

Since the summarization task of DUC 2005 and DUC 2006 are essentially the same, we used the DUC 2005 data to tune our system for 2006. ROUGE-1 and ROUGE-2 metrics (Lin and Hovy, 2003) were used to optimize our system. Language model-based generation probabilities performed consistently better than the cosine measure. Interestingly, bigram language models gave us better ROUGE-2 scores while the unigram language models gave better ROUGE-1 scores. However, we only used the unigram models in our final submission.

There are two parameters to be tuned in our system: λ, the Jelinek-Mercer smoothing parameter; and d, the biased LexRank trade-off between the baseline ranking method and the random walk model based on the link weights. In our experiments on DUC 2005 data, we observed the best ROUGE scores when d was around 0.7. This relatively high value of d makes sense since similarity to the topic statement seems to be more important than the inter-sentence similarities in focused summarization. However, when d is higher than 0.7, too much emphasis on the topic statement causes us to miss the sentences that are related but *indirectly* similar to the topic statement. We also set λ to 0.7 based on our experiments on the 2005 data. We only used the "narrative" field of each topic statement.

After ranking the sentences according to their biased LexRank values, we reranked them using the MMR reranking method (Carbonell and Goldstein, 1998). Then we simply picked the top ranked sentences respecting the 250-word summary limit.

With the parameters tuned on the DUC 2005 data, we ran our biased LexRank system on the DUC 2006 focused summarization task. Among the 34 participants in DUC 2006, our system ranked 11th in overall responsiveness, 9th in ROUGE-2, 7th in ROUGE-SU4, and 11th in the pyramid evaluation. These results are promising but lower than the performance of LexRank in the generic summarization task of DUC 2004. We want to improve our framework by investigating different similarity metrics, smoothing methods and better parameter tuning.

References

Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336.

Güneş Erkan and Dragomir R. Radev. 2004a. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Güneş Erkan and Dragomir R. Radev. 2004b. The University of Michigan at DUC 2004. In *Proceedings of the Document Understanding Conferences*, Boston, MA, May.

Oren Kurland and Lillian Lee. 2005. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR*.

Chin-Yew Lin and E.H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada, May 27 - June 1*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Jahna Otterbacher, Güneş Erkan, and Dragomir Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915–922, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.