Improving the Performance of the Random Walk Model for Answering Complex Questions

Yllias Chali and Shafiq R. Joty
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, Canada, T1K 3M4
{chali,jotys}@cs.uleth.ca

Abstract

We consider the problem of answering complex questions that require inferencing and synthesizing information from multiple documents and can be seen as a kind of topic-oriented, informative multi-document summarization. The stochastic, graph-based method for computing the relative importance of textual units (i.e. sentences) is very successful in generic summarization. In this method, a sentence is encoded as a vector in which each component represents the occurrence frequency (TF*IDF) of a word. However, the major limitation of the TF*IDF approach is that it only retains the frequency of the words and does not take into account the sequence, syntactic and semantic information. In this paper, we study the impact of syntactic and shallow semantic information in the graph-based method for answering complex questions.

1 Introduction

After having made substantial headway in factoid and list questions, researchers have turned their attention to more complex information needs that cannot be answered by simply extracting named entities like persons, organizations, locations, dates, etc. Unlike informationally-simple factoid questions, complex questions often seek multiple different types of information simultaneously and do not presupposed that one single answer could meet all of its information needs. For example, with complex questions like "What are the causes of AIDS?", the wider focus of this question suggests that the submitter may not have a single or well-defined information need and therefore may be amenable to receiving additional supporting information that is relevant to some (as yet) undefined informational goal. This type of questions require inferencing and synthesizing information from multiple documents. In Natural Language Processing (NLP), this information synthesis can be seen as a kind of topic-oriented, informative multi-document summarization, where the goal is to produce a single text as a compressed version of a set of documents with a minimum loss of relevant information.

Recently, the graph-based method (LexRank) is applied successfully to generic, multi-document summarization (Erkan and Radev, 2004). A topic-sensitive LexRank is proposed in (Otterbacher et al., 2005). In this method, a sentence is mapped to a vector in which each element represents the occurrence frequency (TF*IDF) of a word. However, the major limitation of the TF*IDF approach is that it only retains the frequency of the words and does not take into account the sequence, syntactic and semantic information thus cannot distinguish between "The hero killed the villain" and "The villain killed the hero". The task like *answering complex questions* that requires the use of more complex syntactic and semantics, the approaches with only TF*IDF are often inadequate to perform fine-level textual analysis.

In this paper, we extensively study the impact of syntactic and shallow semantic information in measuring similarity between the sentences in the random walk model for answering complex questions. We argue that for this task, similarity measures based on syntactic and semantic information performs better and can be used to characterize the

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 9–12,
Columbus, Ohio, USA, June 2008.

relation between a question and a sentence (answer) in a more effective way than the traditional TF*IDF based similarity measures.

2 Graph-based Random Walk Model for Text Summarization

In (Erkan and Radev, 2004), the concept of graph-based centrality is used to rank a set of sentences, in producing generic multi-document summaries. A similarity graph is produced where each node represents a sentence in the collection and the edges between nodes measure the cosine similarity between the respective pair of sentences. Each sentence is represented as a vector of term specific weights. The term specific weights in the sentence vectors are products of term frequency (tf) and inverse document frequency (idf). The degree of a given node is an indication of how much important the sentence is. To apply LexRank to query-focused context, a topic-sensitive version of LexRank is proposed in (Otterbacher et al., 2005). The score of a sentence is determined by a mixture model:

$$p(s|q) = d \times \frac{rel(s|q)}{\sum_{z \in C} rel(z|q)} + (1-d) \times \sum_{v \in C} \frac{sim(s,v)}{\sum_{z \in C} sim(z,v)} \times p(v|q) \quad (1)$$

Where, $p(s|q)$ is the score of a sentence s given a question q, is determined as the sum of its relevance to the question (i.e. $rel(s|q)$) and the similarity to other sentences in the collection (i.e. $sim(s,v)$). The denominators in both terms are for normalization. C is the set of all sentences in the collection. The value of the parameter d which we call "bias", is a trade-off between two terms in the equation and is set empirically. We claim that for a complex task like answering complex questions where the relatedness between the query sentences and the document sentences is an important factor, the graph-based random walk model of ranking sentences would perform better if we could encode the syntactic and semantic information instead of just the bag of word (i.e. TF*IDF) information in calculating the similarity between sentences. Thus, our mixture model for answering complex questions is:

$$p(s|q) = d \times TREESIM(s,q) + (1-d) \times \sum_{v \in C} TREESIM(s,v) \times p(v|q) \quad (2)$$

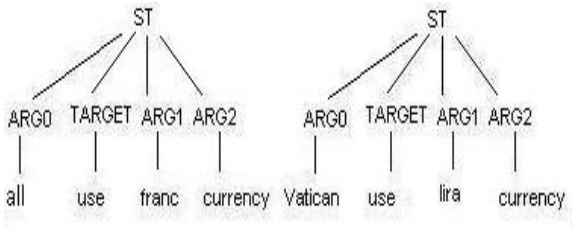

Figure 1: Example of semantic trees

Where *TREESIM(s,q)* is the normalized syntactic (and/or semantic) similarity between the query (q) and the document sentence (s) and C is the set of all sentences in the collection. In cases where the query is composed of two or more sentences, we compute the similarity between the document sentence (s) and each of the query-sentences (q_i) then we take the average of the scores.

3 Encoding Syntactic and Shallow Semantic Structures

Encoding syntactic structure is easier and straight forward. Given a sentence (or query), we first parse it into a syntactic tree using a syntactic parser (i.e. Charniak parser) and then we calculate the similarity between the two trees using the general tree kernel function (Section 4.1).

Initiatives such as PropBank (PB) (Kingsbury and Palmer, 2002) have made possible the design of accurate automatic Semantic Role Labeling (SRL) systems like ASSERT (Hacioglu et al., 2003). For example, consider the PB annotation:

[ARG0 all][TARGET use][ARG1 the french franc][ARG2 as their currency]

Such annotation can be used to design a shallow semantic representation that can be matched against other semantically similar sentences, e.g.

[ARG0 the Vatican][TARGET use][ARG1 the Italian lira][ARG2 as their currency]

In order to calculate the semantic similarity between the sentences, we first represent the annotated sentence using the tree structures like Figure 1 which we call Semantic Tree (ST). In the semantic tree, arguments are replaced with the most important word-often referred to as the semantic head.

The sentences may contain one or more subordinate clauses. For example the sentence, "the Vatican, located wholly within Italy uses the Italian lira

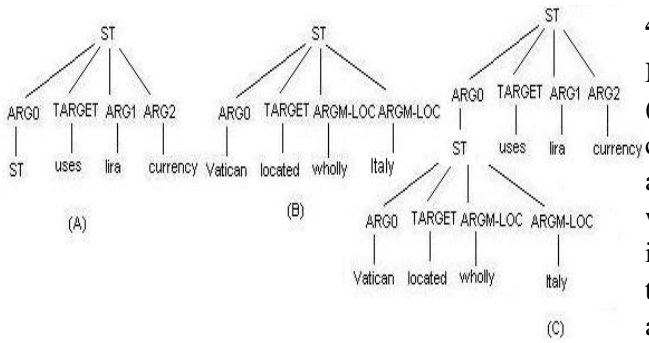

Figure 2: Two STs composing a STN

as their currency." gives the STs as in Figure 2. As we can see in Figure 2(A), when an argument node corresponds to an entire subordinate clause, we label its leaf with ST , e.g. the leaf of ARG0. Such ST node is actually the root of the subordinate clause in Figure 2(B). If taken separately, such STs do not express the whole meaning of the sentence, hence it is more accurate to define a single structure encoding the dependency between the two predicates as in Figure 2(C). We refer to this kind of nested STs as STNs.

4 Syntactic and Semantic Kernels for Text

4.1 Tree Kernels

Once we build the trees (syntactic or semantic), our next task is to measure the similarity between the trees. For this, every tree T is represented by an m dimensional vector $v(T) = (v_1(T), v_2(T), \cdots v_m(T))$, where the i-th element $v_i(T)$ is the number of occurrences of the i-th tree fragment in tree T. The tree fragments of a tree are all of its sub-trees which include at least one production with the restriction that no production rules can be broken into incomplete parts.

Implicitly we enumerate all the possible tree fragments $1, 2, \cdots, m$. These fragments are the axis of this m-dimensional space. Note that this could be done only implicitly, since the number m is extremely large. Because of this, (Collins and Duffy, 2001) defines the tree kernel algorithm whose computational complexity does not depend on m. We followed the similar approach to compute the tree kernel between two syntactic trees.

4.2 Shallow Semantic Tree Kernel (SSTK)

Note that, the tree kernel (TK) function defined in (Collins and Duffy, 2001) computes the number of common subtrees between two trees. Such subtrees are subject to the constraint that their nodes are taken with all or none of the children they have in the original tree. Though, this definition of subtrees makes the TK function appropriate for syntactic trees but at the same time makes it not well suited for the semantic trees (ST) defined in Section 3. For instance, although the two STs of Figure 1 share most of the subtrees rooted in the *ST* node, the kernel defined above computes no match.

The critical aspect of the TK function is that the productions of two evaluated nodes have to be identical to allow the match of further descendants. This means that common substructures cannot be composed by a node with only some of its children as an effective ST representation would require. Moschitti et al. (2007) solve this problem by designing the Shallow Semantic Tree Kernel (SSTK) which allows to match portions of a ST. We followed the similar approach to compute the SSTK.

5 Experiments

5.1 Evaluation Setup

The Document Understanding Conference (DUC) series is run by the National Institute of Standards and Technology (NIST) to further progress in summarization and enable researchers to participate in large-scale experiments. We used the DUC 2007 datasets for evaluation.

We carried out automatic evaluation of our summaries using ROUGE (Lin, 2004) toolkit, which has been widely adopted by DUC for automatic summarization evaluation. It measures summary quality by counting overlapping units such as the n-gram (ROUGE-N), word sequences (ROUGE-L and ROUGE-W) and word pairs (ROUGE-S and ROUGE-SU) between the candidate summary and the reference summary. ROUGE parameters were set as the same as DUC 2007 evaluation setup. All the ROUGE measures were calculated by running ROUGE-1.5.5 with stemming but no removal of stopwords. The ROUGE run-time parameters are:

ROUGE-1.5.5.pl -2 -1 -u -r 1000 -t 0 -n 4 -w 1.2 -m -l 250 -a

The purpose of our experiments is to study the impact of the syntactic and semantic representation for complex question answering task. To accomplish this, we generate summaries for the topics of DUC 2007 by each of our four systems defined as below:

(1) TF*IDF: system is the original topic-sensitive LexRank described in Section 2 that uses the similarity measures based on tf*idf.

(2) SYN: system measures the similarity between the sentences using the *syntactic tree* and the *general tree kernel* function defined in Section 4.1.

(3) SEM: system measures the similarity between the sentences using the *shallow semantic tree* and the *shallow semantic tree kernel* function defined in Section 4.2.

(4) SYNSEM: system measures the similarity between the sentences using both the *syntactic* and *shallow semantic* trees and their associated *kernels*. For each sentence it measures the syntactic and semantic similarity with the query and takes the average of these measures.

5.2 Evaluation Results

The comparison between the systems in terms of their F-scores is given in Table 1. The SYN system improves the ROUGE-1, ROUGE-L and ROUGE-W scores over the TF*IDF system by 2.84%, 0.53% and 2.14% respectively. The SEM system improves the ROUGE-1, ROUGE-L, ROUGE-W, and ROUGE-SU scores over the TF*IDF system by 8.46%, 6.54%, 6.56%, and 11.68%, and over the SYN system by 5.46%, 5.98%, 4.33%, and 12.97% respectively. The SYNSEM system improves the ROUGE-1, ROUGE-L, ROUGE-W, and ROUGE-SU scores over the TF*IDF system by 4.64%, 1.63%, 2.15%, and 4.06%, and over the SYN system by 1.74%, 1.09%, 0%, and 5.26% respectively. The SEM system improves the ROUGE-1, ROUGE-L, ROUGE-W, and ROUGE-SU scores over the SYNSEM system by 3.65%, 4.84%, 4.32%, and 7.33% respectively which indicates that including syntactic feature with the semantic feature degrades the performance.

Systems	ROUGE 1	ROUGE L	ROUGE W	ROUGE SU
TF*IDF	0.359458	0.334882	0.124226	0.130603
SYN	0.369677	0.336673	0.126890	0.129109
SEM	0.389865	0.356792	0.132378	0.145859
SYNSEM	0.376126	0.340330	0.126894	0.135901

Table 1: ROUGE F-scores for different systems

6 Conclusion

In this paper, we have introduced the syntactic and shallow semantic structures and discussed their impacts in measuring the similarity between the sentences in the random walk framework for answering complex questions. Our experiments suggest the following: (a) similarity measures based on the syntactic tree and/or shallow semantic tree outperforms the similarity measures based on the TF*IDF and (b) similarity measures based on the shallow semantic tree performs best for this problem.

References

M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.

G. Erkan and D. R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

K. Hacioglu, S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky. 2003. Shallow Semantic Parsing Using Support Vector Machines. In *Technical Report TR-CSLR-2003-03*, University of Colorado.

P. Kingsbury and M. Palmer. 2002. From Treebank to PropBank. In *Proceedings of the international conference on Language Resources and Evaluation*, Las Palmas, Spain.

C. Y. Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain.

A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classificaion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague, Czech Republic. ACL.

J. Otterbacher, G. Erkan, and D. R. 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, Canada.