Modeling Novelty and Feature Combination using Support Vector Regression for Update Summarization

by

Praveen Bysani, Vijay Bharath Reddy, Vasudeva Varma

in

The 7th International Conference on Natural Language Processing (ICON 2009)

University of Hyderabad

Report No: IIIT/TR/2009/220

Centre for Search and Information Extraction Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2009

Modeling Novelty and Feature Combination using Support Vector Regression for Update Summarization

Praveen Bysani
LTRC
IIIT Hyderabad
lvsnpraveen@research.iiit.ac.in

Vijay Bharat
LTRC
IIIT Hyderabad
yaramvr@research.iiit.ac.in

Vasudeva Varma
LTRC
IIIT Hyderabad
vv@iiit.ac.in

Abstract

Summarization is the process of condensing a piece of text while retaining important information. A well composed and coherent summary is the solution for information overload. Sentence extractive summarization system requires different features to rank sentences and then generate summaries. In this paper we provide a detailed analysis about effect of various features in context of update summarization. We adapt a machine learning algorithm for combining features while scoring a sentence. Further, we propose a new feature that can effectively capture novelty along with relevancy of a sentence in a topic. Evaluation results show that our summmarizer is able to surpass top performing systems participated at Text analysis conference 2008. Gap between oracle summaries and state of art summaries is analyzed to depict the scope of improvement in sentence extractive summarization.

1 Introduction

With the rapid growth of data on the world wide web, it has become important to provide only relevant and useful information to user. Text summarization is introduced as a technique to create compressed version of given text retaining vital information. Types of Summarization varies from "*single document Vs multi-document*", "*query focused Vs query independent*", "*personalized Vs generic*", "*extractive Vs abstractive*".

Recent focus within summarization community has been towards query focused update summarization. The task is to summarize a cluster of documents under the assumption that user had some prior knowledge on topic. The major challenge in update summarization is to detect information that is not only relevant to users need but also novel given the user's prior knowledge. An efficient update summarization system will help user to monitor major changes in a temporally evolving topic especially newswire.

Update summarization shares similarity with Novelty track introduced at TREC 2002[1]. The Novelty track was designed to investigate systems' ability to locate relevant, novel information within the ranked set of documents retrieved in answer to a topic. Researchers have approached the problem of "Update Summarization" at varying levels of complexity. HITIRS (He et al., 2008) proposed an iterative feedback based evolutionary manifold ranking of sentences for update summarization. NUS (ziheng Lin et al., 2008) followed time stamped graph approach incorporating information about temporal ordering of events in articles to focus on the update summary. There are also simple content filtering approaches (Zhang et al., 2008) which identify dynamic content and generate summaries.

Recent advances in machine learning have been adapted to summarization throughout the years. Machine learning models like perceptrons (Fisher and Roark, 2006), markov models (M.Conro et al., 2004), CRF's (shen et al., 2007) and bayesian classifiers (kupeic et al., 1995) have been used throughout the literature for sentence ranking.

We use a machine learning method, Support vector regression (SVR) for sentence ranking. Regression has previously been used for various problems in Information Retrieval and Extraction (Larson, 2002) (Zhang et al., 2003). SVR is very

[1] http://trec.nist.gov/data/novelty.html

Proceedings of ICON-2009: 7th International Conference on Natural Language Processing, Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2009

popular in approximating unknown values from a set of known dependent variables. We consider sentence importance as a unknown value and estimate it using sentence scoring features through Regression. Institute of computational linguistics at Peking university (Li et al., 2007) were the first to use regression in the context of text summarization to predict sentence scores. FastSum (schilder and Kondadandi, 2008) also utilizes support vectors to score sentences using multiple features.

Our work provides an analysis about impact of various features and their combinations on the update summarization. We also propose a new feature Novelty Factor (NF) that models novelty along with relevance in update summarization scenario.

The rest of the paper is organised as follows. In section 2 we briefly describe about support vector regression, estimation of sentence importance and our method of summary generation using SVR. Next in section 3 we explain about features used in sentence ranking and introduce Novelty Factor (NF), then we present our experiments and results of performance of different features in section 4. Finally in section 5 we discuss about the results and provide our analysis.

2 Our Approach

We build a sentence extractive summarizer that extracts and ranks sentences before finally generating summaries. For sentence scoring we utilize a machine learning algorithm, Support Vector Regression (SVR) to predict *sentence rank* using various features(in Section 3). In following sections we briefly explain SVR, estimation of sentence importance and our algorithm to generate summaries.

2.1 Support Vector Regression

Regression analysis refers to techniques for modeling values of a dependent variable from one or more independent variables. Support Vector Machines, a popular mechanism for classification purposes could also be used for regression purposes (Gunn, 1998).

Consider the problem of approximating the set of training data

$$T = \{(F_1, i_1), (F_2, i_2)...(F_s, i_s)\} \subset F \times R.$$

where F is space of feature vectors and R is the set of Real Numbers.

A tuple (F_s, i_s) represents feature vector F_s and importance score i_s of sentence s. Each sample satisfies a linear function $q(f) = \langle w, f \rangle + b$, with $w \in F, b \in R$.

The optimal regression function is given by minimum of fuctional,

$$\Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i{}^- + \xi_i{}^+$$

where C is a pre-specified value, and $\xi_i{}^-, \xi_i{}^+$are slack variables representing upper and lower constraints on the outputs of the system.

We use Radial bias kernel function for our experiments. Like other machine learning algorithms, Support vector regression has two phases, training and testing. During training phase we compute feature vector of each sentence along with its importance. In testing phase, feature vectors of all sentences are generated and their corresponding sentence importance is assessed by trained model.

2.1.1 Sentence Importance Estimation

Importance score (i_s) is not pre-defined for sentences in training data, we estimate the value of importance using human written summaries(also known as models) on that topic.

ROUGE (Lin, 2004) is a recall oriented metric which automatically evaluates machine generated summaries based on their overlap with models. ROUGE-2 and ROUGE-su4 scores highly correlate with human evaluation (Lin and Chin-Yew, 2004). Hence we make an assumption that importance of a sentence is directly proportional to its overlap with model summaries.

Sentence importance is estimated as the ROUGE-2 score of that sentence. The importance of a sentence s, denoted by i_s is computed as follows

$$i_s = \frac{\sum_{m \in models} |Bigram_m \bigcap Bigram_s|}{|s|} \quad (1)$$

$|Bigram_m \bigcap Bigram_s|$ is number of bigrams shared by both model m and sentence s. This count is normalized using sentence length $|s|$.

2.2 Algorithm

Our summarizer follows a 3 stage algorithm to generate summaries,

1. **Pre-Processing**
   In pre-processing stage, documents are

cleaned from news heads and HTML tags. Stop words are removed and porter stemmer is used to derive root words eliminating suffixes and prefixes. Sentences are extracted from each document.

2. **Feature Combination**
   Features used for sentence scoring are combined to rank sentences. Normally features are manually weighted to compute sentence rank. This process is automated with use of SVR in 3 steps,
   - *Sentence tuple generation*: Feature values of every sentence are extracted and its importance(i_s) is estimated as described in Section 2.1.1. Each sentence s in training data is converted into a tuple of form (F_s, i_s). F_s is vector of feature values of sentence $F_s = \{f_1, f_2, f_3\}$
   - *Model building*: A training model is built using SVR, from generated sentence tuples.
   - *sentence scoring*: Importance of a sentence in testing dataset is predicted based on the the trained model. The estimated importance value is considered as rank of sentence.
     $$i_s = q(F_s)$$
3. **Summary Generation**
   During summary generation, a subset of ranked sentences are selected to generate summary. A redundancy check is done between a sentence and summary generated so far, before selecting it into summary. This step helps to prevent duplicate sentences in summary. Sentences are adjusted based on their order of occurence in documents to improve readability. Reported speech is removed to alleviate conciseness of summary.

3 Features

Since we have a machine learning algorithm at our disposal to carry out the tedious job of combining features and scoring sentences, we are able to carry out experiments on various features. Following are the features we used for sentence scoring

3.1 Sentence position

Sentence position is a very old and popular feature used in summarization (Edmundson, 1969). It is well studied and still used as a feature in most state of art summarization systems (Katragadda et al., 2009) (Kastner and Monz, 2009). We use the location information of a sentence in two seperate ways to score a sentence.

Sentence Location 1 (SL1):

First three sentences of a document generally contain the most informative content of that document which is proved by our analysis on the oracle summaries (in Section 4.3). Nearly 40% of all the sentences of the oracle summaries come from among the first three sentences of each document.

Score of a sentence s at position n in document d is given by,

$$\begin{aligned} Score(s_{nd}) &= 1 - \frac{n}{1000} \qquad if \;\; n <= 3 \\ &= \frac{n}{1000} \qquad else \end{aligned}$$

(Assuming that number of sentences in a document will be less than 1000)

Such that,

$$Score(s_{1d}) > Score(s_{2d}) > Score(s_{3d}) >> Score(s_{nd})$$

Sentence Location 2 (SL2):

Positional index of a sentence in the document is assigned as the value of feature. Training model will learn the optimum sentence position for the dataset based on its genre. Hence this feature is not inclined towards top or bottom few sentences in a document like SL1.

$$Score(s_{nd}) = n$$

where s_n is nth sentence in document d.

3.2 Sentence Frequency score (SFS):

Sentence frequency score of a word is defined as the ratio of number of sentences in which the word occurred in document set to the total number of sentences in document set.

sfs score of a word w is given by,

$$sfs(w) = \frac{\{|s| : w \in s\}}{|N|}$$

where s is a sentence, $|N|$ is total number of sentences in dataset.

Average sentence frequency score of words in a sentence is considered as its feature score.

$$Score(s) = \frac{\sum_{i \in s} sfs(w_i)}{|s|}$$

3.3 TF-IDF

TF-IDF is a popular measure in information retrieval to find out relevance of document. Similar analogy is here used to find relevance of a sentence to a document.

Term frequency of a term(t_i) is simply ratio of number of times it occurred in a document(d_j) to total number of terms in document.

$$Tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Inverse Document Frequency of a term(t_i) is ratio of total number of documents in cluster to number of documents in which term occurred.

$$Idf_i = log \frac{|D|}{\{|d| : w \in d\}}$$

Final score of sentence(s) is average Tf-Idf value of terms in it

$$Score(s) = \frac{\sum_{i \in s} Tf_{i,j} * Idf_i}{|s|}$$

3.4 Novelty Factor (NF)

We propose a new feature *novelty factor* (*NF*) that primarily focuses on update summarization problem. Consider a stream of articles published on a topic over time period T. All the articles published from time 0 to time t is assumed to have been read previously (previous clusters). Articles published in the interval t to T are unread articles that might contain new information (new cluster). Let the publishing date of a document d is represented by t_d. *NF* of a word is calculated by

$$nf(w) = \frac{|nd_t|}{|pd_t| + |D|}$$

$nd_t = \{d : w \in d \wedge t_d > t\}$
$pd_t = \{d : w \in d \wedge t_d <= t\}$
$D = \{d : t_d > t\}$

Numerator $|nd_t|$ is the number of documents in the new cluster that contain word w. It is directly proportional to relevancy of the term, since all the documents in the cluster are relevant to the topic. The term $|pd_t|$ in denominator will penalize any word that occurs frequently in previous clusters, in other words it elevates novelty of a term. $|D|$ is total number of documents in current cluster, this is useful for smoothing Novelty Factor when w does not occur in previous clusters.

Score of a sentence s is the average nf value of its content words.

$$Score(s) = \frac{\sum_{i \in s} nf(w_i)}{|s|}$$

NF score of a sentence is a measure of its relevance and novelty to the topic.

3.5 Document Frequency Score (DFS):

Document frequency score (schilder and Kondadandi, 2008) of a word is defined as ratio of number of documents in which it occurred in document set to total number of documents.

dfs score of a word w is given by,

$$dfs(w) = \frac{\{|d| : t_i \in d\}}{|D|}$$

where d is document, $|D|$ is total number of documents in dataset.

Apart from these features, we implemented Probabilistic hyperspace analogue to language (PHAL) (jagadeesh et.al (Jagarlamudi et al., 2006)), KullbackLeibler divergence (KL) (Csiszár and Shields, 2004) as features in our system.

4 Experiments

We used DUC 2007[2] main task documents and corresponding models to generate training data for our experiments. It provides 45 topics, each consisting of 25 documents and a query. Each topic has 4 models summaries of 250 words.

Update summarization task was a pilot task at DUC 2007, its document collection is a subset of main task. It provides 10 topics each divided into 3 clusters A, B, C in chronological order of documents. Cluster A has 10 documents, B has 8, and C has 7 documents respectively. Every cluster has 4 model summaries of 100 words. We used these topics to generate training data for features which are exclusively focused on updating user with new information.

Testing is carried out on TAC 2008 Update summarization[3] dataset. It consists of 48 topics, each topic contains 20 documents divided in chronological order between cluster A and cluster B. Summary for cluster A is normal multi document summary of length 100 words, where as summary for cluster B is an update summary of 100 words.

[2] www-nlpir.nist.gov/projects/duc/guidelines/2007.html
[3] http://www.nist.gov/tac/tracks/2008/summarization/

Experiments are carried out at two levels. First, every feature is individually tested to know their respective performance, and then all combinations of features are tested.

4.1 Individual Features

All the features that are described in Section 3 are evaluated separately to assess their effectiveness at update summarization task. In this level, feature vector (F_s) of sentence s will have only one value. ROUGE scores of every feature are reported in Table1 along with the results of $baseline$ summarizer that generates summary by picking first 100 words of last document in the cluster.

Feature	ROUGE-2	ROUGE-su4
KL	0.09285	0.132325
DFS	0.092225	0.13281
NF	0.086155	0.126455
SL1	0.086245	0.12163
SL2	0.08599	0.12147
SFS	0.077745	0.12419
TF-IDF	0.07317	0.12604
PHAL	0.06505	0.10712
baseline	0.05865	0.09333

Table 1: Average ROUGE-2, ROUGE-su4 recall scores of clusters A, B for individual features

Simple features like DFS, NF, SL1 and SL2 are able to generate very good summaries compared to complex language modeling techniques like PHAL and KL.

4.2 Combinations of Features

All possible combinations of features are evaluated. Feature vector (F_s) of sentence s will have a value corresponding to every feature in combination. We present below the best performing combinations among all.

Combination of DFS + SL2 and NF + SL2 achieves very good results. Surprisingly, KL that best performs at individual level fails to behave the same way when combined with other features. None of the features are complimenting with KL.

4.3 Oracle Summaries

We generated *sentence-extractive Oracle Summaries* using document collection and model summaries. Each oracle summary is the best sentence extractive summary that can be generated by any sentence extractive summarization system for that topic. Sentences are ranked using Equation 1 to produce these summaries.

Combination	ROUGE 2	ROUGE su4
DFS+SL1	0.102195	0.139205
NF+SL1	0.100845	0.13742
DFS+SL2	0.10126	0.13943
NF+SL2	0.0978	0.134925
DFS+TF-IDF	0.0993	0.1383
PHAL+KL	0.094035	0.134275
{DFS+SP +PHAL+KL}	0.09749	0.13705

Table 2: Average ROUGE-2, ROUGE-su4 recall scores of clusters A, B for the best combinations of features

Investigating results of individual clusters will reveal the effect of feature combinations on update summaries. Best configurations for respective clusters are compared against top performing systems at TAC 2008 and the oracle summary in Tables 3 and 4.

System	ROUGE 2	ROUGE su4
DFS+SL2	**0.10604**	**0.13936**
DFS+TF-IDF	**0.10633**	**0.14415**
System-43	0.11137	0.14297
System-13	0.11045	0.13987
Oracle summary	0.17041	0.19616

Table 3: ROUGE-2,ROUGE-su4 recall scores of cluster A

Results of cluster A affirm that combinations of DFS with SL2 and DFS with TF-IDF achieves ample results. These combinations are as good as top performing systems at TAC 2008 .

System	ROUGE 2	ROUGE su4
DFS+SL1	**0.10343**	**0.14267**
NF+SL1	**0.10055**	**0.13791**
System-14	0.10111	0.13669
System-65	0.09675	0.13381
Oracle Summary	0.17610	0.19877

Table 4: ROUGE-2,ROUGE-su4 recall scores of cluster B

Combinations of DFS with SL1 and NF with SL2 clearly outperforms top systems at TAC 2008 in cluster B results. ROUGE-2, ROUGE-su4 scores are improved by approximately 3% over the best summarization system.

5 Discussion

Document frequency score is able to perform well in summarization because all the documents given under a topic are relevant to it. Hence, the importance of a term is directly proportional to the number of documents in which it occurs. While DFS captures relevance of a term, NF reveals the novelty of the term along with its relevance to topic.

NF is the only feature in our current experiments that is specifically tailor made for Update task. It requires a previous set of documents to compute score for a sentence. Training data for NF is scarce compared to other features since we have only DUC 2007 Pilot task data for training. Even this data has not been properly handcrafted for the purpose of update summarization. With a better training dataset, NF is expected to perform better than DFS in update task.

NF has been used for Update summarization task at TAC 2009, using TAC 2008 update data as training set. The 2008 dataset has been created in the purview of update summarization guidelines. Hence it would be a better training set than we use currently. The results of TAC 2009 update summarization task are awaited.

Both sentence positional algorithms (SL1 and SL2) have performed decently. SL1 is inclined towards top sentences in a document, and SL2 is unbiased towards positional index of a sentence. SL1 works because of the intuition that top sentences would always have informative content. SL2 is a feature that helps boosting informative sentences based on genre of the corpus. SL2 learns significant sentence positions in corpus based on training data. As both training and testing belong to same genre (Newswire) SL2 performs well. In our experiments both SL1 and SL2 performs in a similar way as important sentences in training data are present among top 3 sentences of a document. SFS and TF-IDF achieves just about average results in terms of ROUGE.

The huge gap between ROUGE scores of oracle summary and machine generated summaries reveals the scope for improvement in sentence extractive summarization.

6 Conclusion and Future work

Update summarization is a challenging task because the summarized information must be novel apart from being relevant to the user need. In this work, we analyzed significance of various sentence ranking features and their combinations on the update summarization task.

A machine learning approach comes very handy in combining features as it manages the job of tolerating outliers in training data samples created by features that are not completely complementing. We also proposed a new query-independent word level feature, NF, that models novelty and relevance of a sentence in a topic. A combination of the features used in the current system, is able to outperform the top performing systems at TAC 2008 Update Summarization task.

Based on our analysis with the oracle summaries we see that there is a lot of scope for improvement in update summaries. NF, DFS, SL1 and SL2 are all query-independent features that produce generic summaries. In future, we try to incorporate query-focus to our new feature (NF). We also plan to involve NF within a formal language modeling framework. We are currently working on predicting word level importance using SVR. Experiments are being carried out to predict the role of word position in a sentence to decide its importance.

References

Imre Csiszár and Paul C. Shields. 2004. Information theory and statistics: a tutorial. *Commun. Inf. Theory*, 1(4):417–528.

H.P Edmundson. 1969. New methods in automatic extracting. pages 268–285. In Journal of ACM, Volume 16.

Seeger Fisher and Brian Roark. 2006. Query-focused summarization by supervised sentence ranking and skewed word distributions. In *In proceedings of DUC 2006*. DUC.

Steve R. Gunn. 1998. Support vector machines for classification and regression. May.

Ruifang He, Yang Liu, Bing Qin, Ting Liu, and Sheng Li. 2008. Hitirs update summary at tac2008:extractive content selection for language independence. In *TAC 2008 Proceedings*. Text analysis conference, December.

Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma. 2006. Query independent sentence scoring approach to duc 2006. In *In proceedings of DUC 2006*. DUC.

Itamar Kastner and Christof Monz. 2009. Automatic single-document key fact extraction from newswire articles. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*,

pages 415–423, Athens, Greece, March. Association for Computational Linguistics.

Rahul Katragadda, Prasad Pingali, and Vasudeva Varma. 2009. Sentence position revisited: A robust light-weight update summarization baseline algorithm. In *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3)*, pages 46–52, Boulder, Colorado, June. Association for Computational Linguistics.

Julian kupeic, Jan pedersen, and Francine chen. 1995. A trainable document summarizer. In *In proceedings of ACM SIGIR 95*, pages 68–73. ACM.

Ray R. Larson. 2002. A logistic regression approach to distributed ir. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 399–400, New York, NY, USA. ACM.

Sujian Li, You Ouyang, Wei Wang, and Bin Sun. 2007. Multi-document summarization using support vector regression. In *DUC 2007 notebook, 2007*. Document Understanding Conference, November.

Lin and Chin-Yew. 2004. Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough? In Proceedings of the NTCIR Workshop 4, June.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

John M.Conro, Judith D. schlesinger, Jade Goldstein, and Dian P.O'leary. 2004. Left-brain/right-brain multi-document summarization. In *In proceedings of DUC 2004*.

Frank schilder and Ravikumar Kondadandi. 2008. Fastsum: fast and accurate query-based multi-document summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*. Human Language Technology Conference.

Dou shen, Jian-Tao sun, Huan Li, Qiang yang, and Zheng. 2007. Document summarization using conditional random fields. In *In proceedings of IJCAI'07*, pages 2862–2867. IJCAI.

Jian Zhang, Yiming Yang, and Jaime Carbonell. 2003. New event detection with nearest neighbour, support vector machines, and kernel regression. March.

Jin Zhang, Xueqi Cheng, Hongbo Xu, Xiaolei Wang, and Yiling Zeng. 2008. Summarizing dynamic information with signature terms based content filtering. In *TAC 2008 Proceedings*. Text analysis conference, December.

ziheng Lin, Huu Hung Hoang, Long Qiu, Shiren Ye, and Min-Yen Ka. 2008. Nus at tac 2008:augmenting timestamped graphs with event information and selectively expanding opinion contents. In *TAC 2008 Proceedings*. Text analysis conference, December.