Support Vector Machines for Query-focused Summarization trained and evaluated on Pyramid data

Maria Fuentes
TALP Research Center
Universitat Politècnica de Catalunya
mfuentes@lsi.upc.edu

Enrique Alfonseca
Computer Science Departament
Universidad Autónoma de Madrid
Enrique.Alfonseca@gmail.com

Horacio Rodríguez
TALP Research Center
Universitat Politècnica de Catalunya
horacio@lsi.upc.edu

Abstract

This paper presents the use of Support Vector Machines (SVM) to detect relevant information to be included in a query-focused summary. Several SVMs are trained using information from pyramids of summary content units. Their performance is compared with the best performing systems in DUC-2005, using both ROUGE and autoPan, an automatic scoring method for pyramid evaluation.

1 Introduction

Multi-Document Summarization (MDS) is the task of condensing the most relevant information from several documents in a single one. In terms of the DUC contests[1], a query-focused summary has to provide a "brief, well-organized, fluent answer to a need for information", described by a short query (two or three sentences). DUC participants have to synthesize 250-word sized summaries for fifty sets of 25-50 documents in answer to some queries.

In previous DUC contests, from 2001 to 2004, the manual evaluation was based on a comparison with a single human-written model. Much information in the evaluated summaries (both human and automatic) was marked as "related to the topic, but not directly expressed in the model summary". Ideally, this relevant information should be scored during the evaluation. The pyramid method (Nenkova and Passonneau, 2004) addresses the problem by using multiple human summaries to create a gold-standard, and by exploiting the frequency of information in the human summaries in order to assign importance to different facts. However, the pyramid method requires to manually matching fragments of automatic summaries (peers) to the Semantic Content Units (SCUs) in the pyramids. AutoPan (Fuentes et al., 2005), a proposal to automate this matching process, and ROUGE are the evaluation metrics used.

As proposed by Copeck and Szpakowicz (2005), the availability of human-annotated pyramids constitutes a gold-standard that can be exploited in order to train extraction models for the summary automatic construction. This paper describes several models trained from the information in the DUC-2006 manual pyramid annotations using Support Vector Machines (SVM). The evaluation, performed on the DUC-2005 data, has allowed us to discover the best configuration for training the SVMs.

One of the first applications of supervised Machine Learning techniques in summarization was in Single-Document Summarization (Ishikawa et al., 2002). Hirao et al. (2003) used a similar approach for MDS. Fisher and Roark (2006)'s MDS system is based on perceptrons trained on previous DUC data.

2 Approach

Following the work of Hirao et al. (2003) and Kazawa et al. (2002), we propose to train SVMs for ranking the candidate sentences in order of relevance. To create the training corpus, we have used the DUC-2006 dataset, including topic descriptions, document clusters, peer and manual summaries, and pyramid evaluations as annotated during the DUC-2006 manual evaluation. From all these data, a set

[1] http://www-nlpir.nist.gov/projects/duc/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 57–60,
Prague, June 2007.

of relevant sentences is extracted in the following way: first, the sentences in the original documents are matched with the sentences in the summaries (Copeck and Szpakowicz, 2005). Next, all document sentences that matched a summary sentence containing at least one SCU are extracted. Note that the sentences from the original documents that are not extracted in this way could either be positive (i.e. contain relevant data) or negative (i.e. irrelevant for the summary), so they are not yet labeled. Finally, an SVM is trained, as follows, on the annotated data.

Linguistic preprocessing The documents from each cluster are preprocessed using a pipe of general purpose processors performing tokenization, POS tagging, lemmatization, fine grained Named Entities (NE)s Recognition and Classification, anaphora resolution, syntactic parsing, semantic labeling (using WordNet synsets), discourse marker annotation, and semantic analysis. The same tools are used for the linguistic processing of the query. Using these data, a semantic representation of the sentence is produced, that we call *environment*. It is a semantic-network-like representation of the semantic units (nodes) and the semantic relations (edges) holding between them. This representation will be used to compute the (Fuentes et al., 2006) lexico-semantic measures between sentences.

Collection of positive instances As indicated before, every sentence from the original documents matching a summary sentence that contains at least one SCU is considered a positive example. We have used a set of features that can be classified into three groups: those extracted from the sentences, those that capture a similarity metric between the sentence and the topic description (query), and those that try to relate the cohesion between a sentence and all the other sentences in the same document or collection.

The attributes collected **from the sentences** are:

- The position of the sentence in its document.
- The number of sentences in the document.
- The number of sentences in the cluster.
- Three binary attributes indicating whether the sentence contains positive, negative and neutral discourse markers, respectively. For instance, *what's more* is positive, while *for example* and *incidentally* indicate lack of relevance.
- Two binary attributes indicating whether the sentence contains *right-directed* discourse markers (that affect the relevance of fragment after the marker, e.g. *first of all*), or discourse markers affecting both sides, e.g. *that's why*.
- Several boolean features to mark whether the sentence starts with or contains a particular word or part-of-speech tag.
- The total number of NEs included in the sentence, and the number of NEs of each kind.
- *SumBasic score* (Nenkova and Vanderwende, 2005) is originally an iterative procedure that updates word probabilities as sentences are selected for the summary. In our case, word probabilities are estimated either using only the set of words in the current document, or using all the words in the cluster.

The attributes that **depend on the query** are:

- Word-stem overlapping with the query.
- Three boolean features indicating whether the sentence contains a subject, object or indirect object dependency in common with the query.
- Overlapping between the environment predicates in the sentence and those in the query.
- Two similarity metrics calculated by expanding the query words using Google.
- *SumFocus score* (Vanderwende et al., 2006).

The **cohesion-based** attributes [2] are:

- Word-stem overlapping between this sentence and the other sentences in the same document.
- Word-stem overlapping between this sentence and the other sentences in the same cluster.
- Synset overlapping between this sentence and the other sentences in the same document.
- Synset overlapping with other sentences in the same collection.

Model training In order to train a traditional SVM, both positive and negative examples are necessary. From the pyramid data we are able to identify positive examples, but there is not enough evidence to classify the remaining sentences as positive or negative. Although One-Class Support Vector Machine (OSVM) (Manevitz and Yousef, 2001) can learn from just positive examples, according to Yu et al. (2002) they are prone to underfitting and overfitting when data is scant (which happens in

[2]The mean, median, standard deviation and histogram of the overlapping distribution are calculated and included as features.

this case), and a simple iterative procedure called Mapping-Convergence (MC) algorithm can greatly outperform OSVM (see the pseudocode in Figure 1).

Input: positive examples, POS, unlabeled examples U
Output: hypothesis at each iteration $h'_1, h'_2, ..., h'_k$

1. Train h to identify "strong negatives" in U:
   N_1 := examples from U classified as negative by h
   P_1 := examples from U classified as positive by h
2. Set $NEG := \emptyset$ and $i := 1$
3. Loop until $N_i = \emptyset$,
   3.1. $NEG := NEG \cup N_i$
   3.2. Train h'_i from POS and NEG
   3.3. Classify P_i by h'_i:
      N_{i+1} = examples from P_i classified as negative
      P_{i+1} = examples from P_i classified as positive
5. Return $\{h'_1, h'_2, ..., h'_k\}$

Figure 1: Mapping-Convergence algorithm.

The MC starts by identifying a small set of instances that are very dissimilar to the positive examples, called *strong negatives*. Next, at each iteration, a new SVM h'_i is trained using the original positive examples, and the negative examples found so far. The set of negative instances is then extended with the unlabeled instances classified as negative by h'_i.

The following settings have been tried:

- The set of positive examples has been collected either by matching document sentences to peer summary sentences (Copeck and Szpakowicz, 2005) or by matching document sentences to manual summary sentences.
- The initial set of *strong negative* examples for the MC algorithm has been either built automatically as described by Yu et al. (2002), or built by choosing manually, for each cluster, the two or three automatic summaries with lowest manual pyramid scores.
- Several SVM kernel functions have been tried.

For training, there were 6601 sentences from the original documents, out of which around 120 were negative examples and either around 100 or 500 positive examples, depending on whether the document sentences had been matched to the manual or the peer summaries. The rest were initially unlabeled.

Summary generation Given a query and a set of documents, the trained SVMs are used to rank sentences. The top ranked ones are checked to avoid redundancy using a percentage overlapping measure.

3 Evaluation Framework

The SVMs, trained on DUC-2006 data, have been tested on the DUC-2005 corpus, using the 20 clusters manually evaluated with the pyramid method. The sentence features were computed as described before. Finally, the performance of each system has been evaluated automatically using two different measures: ROUGE and autoPan.

ROUGE, the automatic procedure used in DUC, is based on n-gram co-occurrences. Both ROUGE-2 (henceforward R-2) and ROUGE-SU4 (R-SU4) has been used to rank automatic summaries.

AutoPan is a procedure for automatically matching fragments of text summaries to SCUs in pyramids, in the following way: first, the text in the SCU label and all its contributors is stemmed and stop words are removed, obtaining a set of stem vectors for each SCU. The system summary text is also stemmed and freed from stop words. Next, a search for non-overlapping windows of text which can match SCUs is carried. Each match is scored taking into account the score of the SCU as well as the number of matching stems. The solution which globally maximizes the sum of scores of all matches is found using dynamic programming techniques.

According to Fuentes et al. (2005), autoPan scores are highly correlated to the manual pyramid scores. Furthermore, autoPan also correlates well with manual responsiveness and both ROUGE metrics.[3]

3.1 Results

Positive	Strong neg.	R-2	R-SU4	autoPan
peer	pyramid scores	**0.071**	**0.131**	**0.072**
	(Yu et al., 2002)	0.036	0.089	0.024
manual	pyramid scores	0.025	0.075	0.024
	(Yu et al., 2002)	0.018	0.063	0.009

Table 1: ROUGE and autoPan results using different SVMs.

Table 1 shows the results obtained, from which some trends can be found: firstly, the SVMs trained using the set of positive examples obtained from peer summaries consistently outperform SVMs trained using the examples obtained from the manual summaries. This may be due to the fact that the

[3] In DUC-2005 pyramids were created using 7 manual summaries, while in DUC-2006 only 4 were used. For that reason, better correlations are obtained in DUC-2005 data.

number of positive examples is much higher in the first case (on average 48,9 vs. 12,75 examples per cluster). Secondly, generating automatically a set with seed negative examples for the M-C algorithm, as indicated by Yu et al. (2002), usually performs worse than choosing the strong negative examples from the SCU annotation. This may be due to the fact that its quality is better, even though the amount of seed negative examples is one order of magnitude smaller in this case (11.9 examples in average). Finally, the best results are obtained when using a RBF kernel, while previous summarization work (Hirao et al., 2003) uses polynomial kernels.

The proposed system attains an autoPan value of 0.072, while the best DUC-2005 one (Daumé III and Marcu, 2005) obtains an autoPan of 0.081. The difference is not statistically significant. (Daumé III and Marcu, 2005) system also scored highest in responsiveness (manually evaluated at NIST).

However, concerning ROUGE measures, the best participant (Ye et al., 2005) has an R-2 score of 0.078 (confidence interval [0.073–0.080]) and an R-SU4 score of 0.139 [0.135–0.142], when evaluated on the 20 clusters used here. The proposed system again is comparable to the best system in DUC-2005 in terms of responsiveness, Daumé III and Marcu (2005)'s R-2 score was 0.071 [0.067–0.074] and R-SU4 was 0.126 [0.123–0.129] and it is better than the DUC-2005 Fisher and Roark supervised approach with an R-2 of 0.066 and an R-SU4 of 0.122.

4 Conclusions and future work

The pyramid annotations are a valuable source of information for training automatically text summarization systems using Machine Learning techniques. We explore different possibilities for applying them in training SVMs to rank sentences in order of relevance to the query. Structural, cohesion-based and query-dependent features are used for training.

The experiments have provided some insights on which can be the best way to exploit the annotations. Obtaining the positive examples from the annotations of the peer summaries is probably better because most of the peer systems are extract-based, while the manual ones are abstract-based. Also, using a very small set of strong negative example seeds seems to perform better than choosing them automatically with Yu et al. (2002)'s procedure.

In the future we plan to include features from adjacent sentences (Fisher and Roark, 2006) and use rouge scores to initially select negative examples.

Acknowledgments

Work partially funded by the CHIL project, IST-2004506969.

References

T. Copeck and S. Szpakowicz. 2005. Leveraging pyramids. In *Proc. DUC-2005*, Vancouver, Canada.

Hal Daumé III and Daniel Marcu. 2005. Bayesian summarization at DUC and a suggestion for extrinsic evaluation. In *Proc. DUC-2005*, Vancouver, Canada.

S. Fisher and B. Roark. 2006. Query-focused summarization by supervised sentence ranking and skewed word distributions. In *Proc. DUC-2006*, New York, USA.

M. Fuentes, E. Gonzàlez, D. Ferrés, and H. Rodríguez. 2005. QASUM-TALP at DUC 2005 automatically evaluated with the pyramid based metric autopan. In *Proc. DUC-2005*.

M. Fuentes, H. Rodríguez, J. Turmo, and D. Ferrés. 2006. FEMsum at DUC 2006: Semantic-based approach integrated in a flexible eclectic multitask summarizer architecture. In *Proc. DUC-2006*, New York, USA.

T. Hirao, J. Suzuki, H. Isozaki, and E. Maeda. 2003. Ntt's multiple document summarization system for DUC2003. In *Proc. DUC-2003*.

K. Ishikawa, S. Ando, S. Doi, and A. Okumura. 2002. Trainable automatic text summarization using segmentation of sentence. In *Proc. 2002 NTCIR 3 TSC workshop*.

H. Kazawa, T. Hirao, and E. Maeda. 2002. Ranking SVM and its application to sentence selection. In *Proc. 2002 Workshop on Information-Based Induction Science (IBIS-2002)*.

L.M. Manevitz and M. Yousef. 2001. One-class SVM for document classification. *Journal of Machine Learning Research*.

A. Nenkova and R. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proc. HLT/NAACL 2004*, Boston, USA.

A. Nenkova and L. Vanderwende. 2005. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research.

L. Vanderwende, H. Suzuki, and C. Brockett. 2006. Microsoft research at DUC 2006: Task-focused summarization with sentence simplification and lexical expansion. In *Proc. DUC-2006*, New York, USA.

S. Ye, L. Qiu, and T.S. Chua. 2005. NUS at DUC 2005: Understanding documents via concept links. In *Proc. DUC-2005*.

H. Yu, J. Han, and K. C-C. Chang. 2002. PEBL: Positive example-based learning for web page classification using SVM. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD02)*, New York.